



Louisiana Computer Science: 1st Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Louisiana Computer Science: 1st Grade 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 26 required class sessions across five standards-aligned units, plus optional lessons in math and science, for a total of roughly 39 class sessions of 45 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.

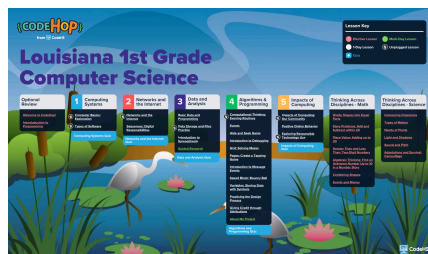
Standards Alignment Note: Lessons that address a specific standard list it in bold in parentheses after the lesson title. Lessons without a standards tag support spiral review, practice, or enrichment.

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 3 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/29032/overview>.



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

Course Breakdown

Optional Review

Optional introductory lessons that help students get comfortable logging in, moving around the CodeHopJr Playground, and building a first scene. These can be taught on their 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 • Create a scene with characters
Lessons	Welcome to CodeHop! • Log in and explore the CodeHopJr Playground before beginning the course. Introduction to Programming • Navigate the programming interface to build a scene with characters.

Unit 1: Computing Systems (2 lessons)

Students take a closer look at what a computer is and how its parts work together. They identify hardware and software, compare software applications, and work out what to do when something does not work.

Objectives / Topics Covered	• Explain how common hardware pieces function • Describe the function of common software components • Match hardware and software to a task • Identify a basic software problem
Lessons	Computer Basics: Exploration (E.CS.1A, E.CS.1C, E.CS.2A) • Identify input, output, hardware, and software, and troubleshoot a device that is not working. Types of Software (E.CS.1B) • Compare software applications and choose the right one for a task. Computing Systems Quiz (3 questions)

Unit 2: Networks and the Internet (2 lessons)

Students explain what a network is and model how messages travel across the Internet. They also program characters to show what responsible online behavior looks like.

Objectives / Topics Covered	• Explain what a network is • Compare wireless and plugged-in connections • Define personally identifiable information • Program a sequence showing responsible online choices
Lessons	Networks and the Internet (E.NI.1A) • Model how messages travel between people over a network and the Internet. Sequences: Digital Responsibilities (E.NI.2A, E.NI.2B) • Program characters with sequences to show responsible online behavior. Networks and the Internet Quiz (3 questions)

Unit 3: Data and Analysis (5 lessons)

Students collect data and present it visually, both in their own programs and in a spreadsheet. They also practice saving and naming files, and find information through guided research.

Objectives / Topics Covered	- Collect data and present it visually - Use spreadsheet software to organize data - Save, name, and reopen a data file - Find information using research sources
Lessons	Basic Data and Programming (E.DA.1B, E.DA.2A, E.DA.2B, E.DA.4B, E.DA.5B) - Collect data and build a program that presents it visually. Data Storage and Files Practice (E.DA.3A, E.DA.3B) - Save, name, and reopen files to model how computers store data. Introduction to Spreadsheets (E.DA.4A) - Use spreadsheet software to organize and present data on class pets and favorite seasons. Guided Research (2 classes E.DA.1A) - Find information using research sources and build a program to communicate it. Data and Analysis Quiz (3 questions)

Unit 4: Algorithms and Programming (14 lessons)

Students build the core programming skills of the course, working with events, messages, sequences, grids, pages, and variables. They also practice debugging, follow the design process, and learn to credit the work of others.

Objectives / Topics Covered	- Program with events, messages, and sequences - Find and fix bugs in a program - Use the design process to plan and revise a project - Give credit when reusing someone else's work
Lessons	Computational Thinking: Evening Routines (E.AP.3A) - Break an evening routine into ordered steps and spot the patterns inside it. Events - Use multiple event blocks in one program and explain what an event does. Hide and Seek Game - Program an interactive game of hide-and-seek using the hide block. Introduction to Debugging (E.AP.4B) - Find and correct bugs in a sequence. Grid: Solving Mazes (E.AP.2A) - Design a maze and program a character to move through it on the grid. Pages: Create a Tapping Game (E.AP.1B) - Build a game that moves from one page to the next using go-to-page blocks. Introduction to Message Events - Program a relay race in which messages make characters interact. Speed Block: Bouncy Ball (E.AP.3B) - Combine speed blocks and messages to change how a character moves. Variables: Storing Data with Symbols (E.AP.1A) - Model how programs store and change data using variables. Practicing the Design Process (E.AP.4A, E.AP.4E, E.AP.4F) - Plan, create, and revise a project that models a solution to a simple real-world problem. Giving Credit Through Attributions (E.AP.4D) - Reuse part of a story to make a new one and give credit to the original creator. About Me Project (3 classes E.AP.4C) - Plan and build a program describing their own characteristics and interests. Algorithms and Programming Quiz (3 questions)

Unit 5: Impacts of Computing (3 lessons)

Students look at how computing shapes their community and what responsible technology use looks like. They practice telling appropriate online behavior from inappropriate behavior and learn the rules that keep them safe.

Objectives / Topics Covered	- Identify computing devices in the community - Explain how technology has changed how people live and work - Tell appropriate from inappropriate online behavior - Describe safe and responsible technology use
Lessons	Impacts of Computing: Our Community (E.IC.1A) - Identify computing devices in the community and how they change the way people live and work. Positive Online Behavior (E.IC.2A, E.IC.2B, E.IC.3A) - Tell appropriate online behavior from inappropriate behavior in a CodeHopJr program. Exploring Responsible Technology Use (E.IC.3B, E.IC.3C) - Explain how to use technology safely and follow school technology rules. Impacts of Computing Quiz (3 questions)

Thinking Across Disciplines - Math (Optional)

Optional lessons that apply coding to first grade math content, programming animations and interactive scenes that model equal parts, place value, comparison, algebraic thinking, and money. These can be taught in any order and integrated throughout the year as needed.

Objectives / Topics Covered	- Partition shapes into equal parts - Model addition and subtraction story problems - Decompose two-digit numbers into tens and ones - Compare two-digit numbers using events
Lessons	Divide Shapes into Equal Parts (E.DA.5A) - Animate circles and rectangles being partitioned into equal parts. Story Problems: Add and Subtract within 20 - Use events to build a scene representing an addition or subtraction story problem. Place Value: Adding Up to 20 - Use events to show how a two-digit number breaks apart into tens and ones. Greater Than and Less Than: Two-Digit Numbers - Program a comparison of two numbers using events. Algebraic Thinking: Find an Unknown Number Up to 10 in a Number Story - Build an interactive program that solves for an unknown number in a number story. Combining Shapes - Create composite shapes, then animate them with event and motion blocks. Events and Money - Program a scene that tells the value of a set of coins.

Thinking Across Disciplines - Science (Optional)

Optional lessons that use programming to explore first grade science concepts, including organism traits, motion, plant needs, light and shadow, sound, and camouflage. These can be taught in any order and integrated throughout the year as needed.

Objectives / Topics Covered	- Group organisms by shared characteristics - Program animations showing types of motion - Model what plants need to survive - Explain the relationship between sound and vibration
Lessons	Comparing Organisms - Build a program that groups animals by shared characteristics. Types of Motion - Use events to animate different types of motion. Needs of Plants

	• Use events and messages to model what plants need to survive. Light and Shadows • Program events that show the sources and effects of light. Sound and Pitch • Animate sound and pitch to explain how sound relates to vibration. Adaptations and Survival: Camouflage • Illustrate and explain how living things use camouflage to survive.
--	--

Louisiana Computer Science: 1st Grade 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 .	