



Louisiana Computer Science: 4th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Louisiana Computer Science: 4th 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 28 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.

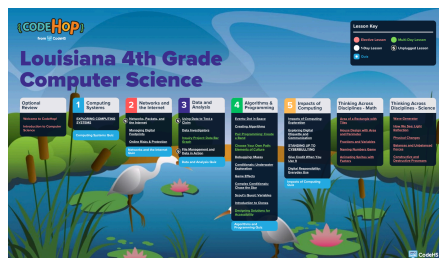
Standards Alignment Note: Lessons that address a specific standard list it in bold in parentheses after the lesson title. Lessons without a standards tag support spiral review, practice, or enrichment.

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 5 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/28980/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 CodeHop Playground, and writing a first program. 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 CodeHop Playground • Define key computer science vocabulary
Lessons	Welcome to CodeHop! • Log in and explore the CodeHop Playground before beginning the course. Introduction to Computer Science • Define important computer science vocabulary and build a simple program.

Unit 1: Computing Systems (1 lesson)

Students identify the parts of a computing system and how hardware and software work together. They also practice spotting simple hardware and software problems.

Objectives / Topics Covered	• Identify the parts of a computing system • Compare the functions of hardware and software • Match hardware and software to a task • Identify simple hardware and software problems
Lessons	Exploring Computing Systems (E.CS.1A, E.CS.1B, E.CS.1C, E.CS.2A) • Identify parts of a computing system and spot simple hardware and software problems. Computing Systems Quiz (5 questions)

Unit 2: Networks and the Internet (3 lessons)

Students compare wired and wireless networks and model how packets carry information between devices. They also examine digital footprints and practice habits that protect their information online.

Objectives / Topics Covered	• Compare wired and wireless network connections • Model how devices transfer information using packets • Explain how online actions create a digital footprint • Practice secure habits that protect personal information
Lessons	Networks, Packets, and the Internet (E.NI.1A) • Compare types of network connections and model how devices transfer information using rules and packets. Managing Digital Footprints (E.NI.2A) • Explain how online actions create permanent digital footprints and how to manage a digital identity. Online Risks & Protection (E.NI.2B) • Practice secure habits and identify the tools that protect personal information online. Networks and the Internet Quiz (5 questions)

Unit 3: Data and Analysis (5 lessons)

Students test claims against data, evaluate sources for reliability, and run an inquiry project that graphs their own investigation. They also compare how different kinds of digital data are sized and stored.

Objectives / Topics Covered	• Use data to test a claim • Evaluate data for reliability • Follow the inquiry process to investigate a question • Compare how digital data is sized and stored
Lessons	Using Data to Test a Claim (E.DA.1B) • Gather and weigh data to decide whether it supports a given claim. Data Investigators (E.DA.4B, E.DA.5A, E.DA.5B) • Evaluate data for reliability, then analyze it to draw conclusions and make predictions. Inquiry Project: Data Bar Graph (2 classes E.DA.1A, E.DA.2A, E.DA.2B, E.DA.4A) • Follow the inquiry process and modify a program to display investigation results as a bar graph. File Management and Data in Action (E.DA.3A, E.DA.3B) • Explain how different types of digital data vary in size and can be stored in different ways. Data and Analysis Quiz (5 questions)

Unit 4: Algorithms and Programming (14 lessons)

Students compare algorithms, program with conditionals, variables, and clones, and collaborate through pair programming. They close the unit by applying design thinking to make a game more accessible.

Objectives / Topics Covered	• Program and compare multiple algorithms • Use conditionals, variables, and clones in a program • Collaborate through pair programming and peer feedback • Apply design thinking to improve accessibility
Lessons	Events: Dot in Space • Build and explain a program that uses multiple types of event blocks. Creating Algorithms (E.AP.1B) • Program multiple algorithms and assess which one best meets a need. Pair Programming: Create a Band (2 classes) • Collaborate through pair programming to design and code a band using keyboard inputs. Choose Your Own Path: Elements of Culture (2 classes E.AP.4D) • Identify elements of culture while building a choose-your-own-path game. Debugging: Mazes (E.AP.3A, E.AP.4B) • Decompose a program to debug it and make it run as intended. Conditionals: Underwater Exploration (E.AP.2A) • Build a program that uses conditionals to drive an underwater scene. Game Effects • Modify a game to add effects and update it based on peer feedback. Complex Conditionals: Chase the Star • Explain what an if/then/else conditional is and use it in a program. Scout's Quest: Variables (E.AP.1A) • Create and use variables to track points in a program. Introduction to Clones • Build an animation using clones and investigate the limitations of the program. Designing Solutions for Accessibility (2 classes E.AP.3B, E.AP.4A, E.AP.4C, E.AP.4E, E.AP.4F) • Use design thinking to redesign a game for better accessibility and usability. Algorithms and Programming Quiz (5 questions)

Unit 5: Impacts of Computing (5 lessons)

Students investigate how technology reshapes jobs, culture, and the environment, and what responsible participation looks like online. They practice digital etiquette, respond to cyberbullying, and attribute sources properly.

Objectives / Topics Covered	• Explain how technology and culture influence each other • Demonstrate proper digital etiquette online • Respond to cyberbullying appropriately • Apply responsible technology rules
Lessons	Impacts of Computing: Exploration (E.IC.1A) • Investigate how technology changes jobs, culture, and the environment. Exploring Digital Etiquette and Communication (E.IC.2A) • Demonstrate proper digital etiquette when communicating in an online community. Standing Up to Cyberbullying (E.IC.2B) • Recognize types of hurtful online behavior and describe ways to respond or take responsibility. Give Credit When You Use It (E.IC.3A) • Search for information online and provide proper attribution to sources. Digital Responsibility: Everyday Use (E.IC.3B, E.IC.3C) • Apply responsible technology rules and describe how those choices affect others. Impacts of Computing Quiz (5 questions)

Thinking Across Disciplines - Math (Optional)

Optional lessons that apply coding to fourth grade math content, programming games and animations that model area and perimeter, fractions, number names, and factors. These can be taught in any order and integrated throughout the year as needed.

Objectives / Topics Covered	• Model area and perimeter using variables and loops • Represent fractions on a number line • Read and write numbers in expanded form • Animate sprites using factors
Lessons	Area of a Rectangle with Tiles • Use variables and loops to model the fractional area of a rectangle with tiles. House Design with Area and Perimeter (2 classes) • Calculate area and perimeter to design a house using functions. Fractions and Variables • Represent fractions on a number line using variables and conditionals. Naming Numbers Game • Build an interactive game using variables and operators to read and write numbers in expanded form. Animating Sprites with Factors • Use factors with loops and wait blocks to animate sprites.

Thinking Across Disciplines - Science (Optional)

Optional lessons that use programming to explore fourth grade science concepts, including waves, light reflection, states of matter, forces, and Earth's changing surface. These can be taught in any order and integrated throughout the year as needed.

Objectives / Topics Covered	• Draw wave patterns using variables and loops • Model how light reflects and enters the eye • Program changes between solid, liquid, and gas • Model balanced and unbalanced forces
-----------------------------------	---

Lessons	Wave Generator - Use variables and loops to draw a variety of wave patterns. How We See: Light Reflection - Use conditionals to show how light reflects from objects and enters the eye. Physical Changes - Program a sprite that changes between solid, liquid, and gas as the user adjusts temperature. Balanced and Unbalanced Forces - Model how balanced and unbalanced forces affect an object's speed using conditionals and variables. Constructive and Destructive Processes - Animate how volcanoes change surface features through a constructive process.
---------	---

Louisiana Computer Science: 4th 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.
Design-Your-Own-Lesson Templates	Empower your students to explore and express their knowledge creatively with our versatile graphic organizer templates. Designed with adaptability and ease of use in mind, these interactive tools transform any subject into an engaging, hands-on learning experience.
These resources and more are found on the CodeHop Resources Page .	