



Louisiana Computer Science: 5th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Louisiana Computer Science: 5th 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 38 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/28981/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	Practicing with 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 (4 lessons)

Students compare wired and wireless networks and model how packets move information between devices. They also analyze how online actions shape a digital reputation and build a program that solves a real cybersecurity problem.

Objectives / Topics Covered	• Compare wired and wireless network components • Model how devices transfer information using packets • Analyze how online actions affect digital reputation • Explain how to protect personal data and digital security
Lessons	Network Connections (E.NI.1A) • Compare network types and model how devices communicate using rules and packets. Digital Reputation and Identity (E.NI.2A) • Analyze how online actions affect digital reputation and how to protect personal data. Cybersecurity & Privacy (2 classes E.NI.2B) • Build an animated program demonstrating how to solve a real-world cybersecurity issue. Networks and the Internet Quiz (5 questions)

Unit 3: Data and Analysis (6 lessons)

Students convert tables into line graphs, analyze data to draw conclusions, and run an inquiry project on a question of their own. They also compare how different kinds of digital data are sized and stored.

Objectives / Topics Covered	• Convert a table of values into a line graph • Analyze data to draw conclusions and make predictions • Follow the inquiry process to investigate a question • Compare how digital data is sized and stored
Lessons	Data About Data (E.DA.1B) • Examine what information a data set carries beyond its values. Using Digital Tools to Create Line Graphs (E.DA.2B) • Convert a table of information into a line graph that supports a claim. Data Analysts (E.DA.4A, E.DA.4B, E.DA.5B) • Analyze data to draw conclusions and make predictions. Inquiry Project: Line Graph (2 classes E.DA.1A, E.DA.2A, E.DA.5A) • Follow the inquiry process and modify a program to display investigation results as a line graph. File Management and Data Exploration (E.DA.3A, E.DA.3B) • Explain how different types of digital data vary in size and identify where each can be stored. Data and Analysis Quiz (5 questions)

Unit 4: Algorithms and Programming (11 lessons)

Students compare and refine algorithms, plan with pseudocode, and program with conditionals, comparison operators, and variables. Two multi-class projects close the unit: designing an app and building a collaborative game.

Objectives / Topics Covered	• Compare and refine algorithms for the same task • Plan a program using pseudocode • Use conditionals and comparison operators in a program • Apply design thinking to build an app or game
Lessons	Compare and Refine Algorithms (E.AP.1B) • Compare multiple algorithms for one task and determine which is most efficient. Debugging: Make a Pizza (E.AP.4B) • Decompose a program to debug it and make it run as intended. Conditionals: Mazes (E.AP.2A) • Plan with pseudocode, then build a maze program that uses conditionals. Game Mechanics with Comparison Operators (E.AP.1A) • Use comparison operators and variables to create ending game mechanics. Design an App (3 classes E.AP.3A, E.AP.4A, E.AP.4E, E.AP.4F) • Use the design thinking process to design an app that solves a user's needs. Giving Credit Through Attributions (E.AP.4D) • Give appropriate attribution when creating or remixing programs online. Game Design Project (3 classes E.AP.3B, E.AP.4C) • Collaborate to design and build a game using loops, conditionals, and variables. Algorithms and Programming Quiz (5 questions)

Unit 5: Impacts of Computing (6 lessons)

Students analyze how broader participation has changed computing and what responsible digital citizenship requires. They research a state cybersecurity law, practice reporting cyberbullying, and collaborate globally to improve a program.

Objectives / Topics Covered	- Analyze how inclusion has changed computing technologies - Demonstrate proper digital etiquette and teamwork - Report cyberbullying and act as an upstander - Explain cybersecurity policies and laws
Lessons	Technology for Everyone (E.IC.1A) - Analyze how including more people has changed computing technologies. Digital Etiquette and Teamwork (E.IC.2A) - Demonstrate proper digital etiquette when communicating in an online community. Reporting Cyberbullying (E.IC.2B) - Compare types of cyberbullying and demonstrate how to report it and act as an upstander. Cybersecurity Policies and Laws (E.IC.3A) - Explain school computing policies and research a cybersecurity law specific to the state. Collaborating Globally (2 classes E.IC.3B, E.IC.3C) - Collaborate with others digitally to improve a program. Impacts of Computing Quiz (5 questions)

Thinking Across Disciplines - Math (Optional)

Optional lessons that apply coding to fifth grade math content, programming calculators, quiz games, and classification tools that model unit conversion, decimal division, rates, and quadrilateral hierarchies. These can be taught in any order and integrated throughout the year as needed.

Objectives / Topics Covered	- Program a unit conversion calculator - Build a quiz game using variables and conditionals - Solve decimal division problems in a program - Classify quadrilaterals in a hierarchy
Lessons	Unit Converter - Use variables and conditionals to program a calculator that converts between measurements. Multiplication Quiz Game - Build a multiplication quiz game using variables, operators, conditionals, and loops. Decimal Division and Conditionals - Solve decimal division problems and program an interactive division game with levels. Rates and Unit Rates - Build a unit rate converter using variables and operators. Classifying Quadrilaterals in a Hierarchy - Use events to classify quadrilaterals in a hierarchy and document the code with comments.

Thinking Across Disciplines - Science (Optional)

Optional lessons that use programming to explore fifth grade science concepts, including Earth's rotation, the cycle of matter, Earth systems, ecosystems, and pollution. These can be taught in any order and integrated throughout the year as needed.

Objectives / Topics Covered	- Model how Earth's rotation causes day and night - Animate the cycle of matter using events and messages - Model how wind and rain affect the geosphere - Simulate how changes affect an ecosystem
Lessons	Day and Night Use comparison operators and conditionals to show how Earth's rotation causes day

	and night. Cycle of Matter • Use events and messages to build an animated model of the cycle of matter. Earth Systems • Program a model showing how wind and rain impact the geosphere. Balanced Ecosystems • Simulate an ecosystem and change its components to show how populations are affected. Effects of Pollution • Write a program with if/then/else blocks explaining how actions affect water pollution.
--	---

Louisiana Computer Science: 5th 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 .	