



Louisiana Computer Science: 2nd Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Louisiana Computer Science: 2nd 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 27 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.

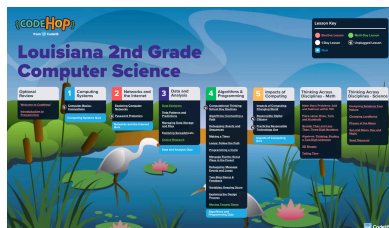
Standards Alignment Note: Lessons that address a specific standard list it in bold in parentheses after the lesson title. Lessons without a standards tag support spiral review, practice, or enrichment.

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 4 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/29033/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 (1 lesson)

Students examine how the parts of a computer work together. They compare types of hardware and software, describe how each contributes to a task, and tell a hardware problem apart from a software problem.

Objectives / Topics Covered	• Compare the functions and uses of various hardware • Compare the functions and uses of various software • Describe how hardware and software complete a task • Tell a hardware problem from a software problem
Lessons	Computer Basics: Connections (E.CS.1A, E.CS.1B, E.CS.1C, E.CS.2A) • Identify input, output, hardware, and software, and explain how they work together. Computing Systems Quiz (4 questions)

Unit 2: Networks and the Internet (2 lessons)

Students model how networks break down and share information between devices. They also learn why usernames and passwords matter and practice strategies for keeping login information safe.

Objectives / Topics Covered	• Describe how networks connect devices to share information • Model how computers break down data to send it • Weigh the risks and benefits of sharing personal information • Identify secure and insecure networks
Lessons	Exploring Computer Networks (E.NI.1A) • Model the sending and receiving of information across a network using message blocks. Password Protectors (E.NI.2A, E.NI.2B) • Practice strategies for keeping usernames and passwords safe. Networks and the Internet Quiz (4 questions)

Unit 3: Data and Analysis (7 lessons)

Students run their own surveys, look for patterns in the results, and present findings both in programs and in spreadsheets. They also manage stored files and assess sources during a research project.

Objectives	• Develop an investigative question and collect survey data
------------	---

/ Topics Covered	- Identify patterns and make predictions from data - Build bar graphs using spreadsheet software - Assess sources to answer a research question
Lessons	Data Explorers (2 classes E.DA.1B, E.DA.5A, E.DA.5B) - Develop an investigative question, run a survey, and program a visual presentation of the results. Data Patterns and Predictions - Use events to communicate patterns, trends, and predictions drawn from a data set. Managing Data Storage and Files (E.DA.3A, E.DA.3B) - Model how computers store data as files, including recovering one that was deleted. Exploring Spreadsheets (E.DA.1A, E.DA.2B, E.DA.4A, E.DA.4B) - Use spreadsheet software to organize and present data on class pets and favorite colors. Choice Research (2 classes E.DA.2A, E.AP.4D) - Collect and assess sources to answer a research question, then communicate the findings visually. Data and Analysis Quiz (4 questions)

Unit 4: Algorithms and Programming (14 lessons)

Students work with loops, message events, variables, and pages to build increasingly complex programs. They debug provided code, give and act on peer feedback, and follow the design process through a multi-class game project.

Objectives / Topics Covered	- Create and refine algorithms to complete a task - Program with loops and message events - Find and fix errors in provided code - Use the design process and peer feedback to improve a program
Lessons	Computational Thinking: School Day Routines (E.AP.4A) - Break a school day routine into ordered steps and simplify the process. Algorithms: Connecting a Path (E.AP.1B) - Create and adjust algorithms that move characters based on size, shape, and starting position. Debugging: Events and Sequences (E.AP.3A) - Find and fix errors in provided code. Making a Timer - Use loops, wait blocks, and turn blocks to build and compare two timers running at different speeds. Loops: Follow the Path (E.AP.2A) - Spot a repeating pattern and use loops to program a path. Programming a Cycle - Use message events to model a repeating cycle. Message Events: Scout Plays in the Forest - Use message events to control the flow of a program. Debugging: Message Events and Loops (E.AP.4B) - Find and fix message event and loop errors in provided code. Two-Step Dance & Feedback (E.AP.4C, E.AP.4E, E.AP.4F) - Revise a program based on peer feedback and give attribution to the peer who helped. Variables: Keeping Score (E.AP.1A) - Build a program that simulates keeping score using a variable. Exploring the Design Process (E.AP.3B) - Plan, build, and improve a program with loops that models a solution to a real-world problem. Moving Targets Game (3 classes) - Combine sequences, events, and pages to build a moving target game.

	Algorithms and Programming Quiz (4 questions)
--	--

Unit 5: Impacts of Computing (3 lessons)

Students compare doing tasks with and without computing systems and consider what responsible digital citizenship means. They discuss digital footprints, cyberbullying, and how to report concerns.

Objectives / Topics Covered	• Compare tasks done with and without computing systems • Explain what a digital footprint is • Recognize cyberbullying and know how to report it • Categorize safe and unsafe uses of technology
Lessons	Impacts of Computing: Changing World (E.IC.1A) • Use programming to explain how technology has changed the way people live, work, and play. Responsible Digital Citizens (E.IC.2A, E.IC.2B, E.IC.3A) • Explain digital footprints, discuss cyberbullying, and practice reporting concerns. Practicing Responsible Technology Use (E.IC.3B, E.IC.3C) • Demonstrate safe and responsible technology use under the school's rules. Impacts of Computing Quiz (4 questions)

Thinking Across Disciplines - Math (Optional)

Optional lessons that apply coding to second grade math content, programming interactive scenes that model story problems, place value, comparison, algebraic thinking, 3D shapes, and telling time. These can be taught in any order and integrated throughout the year as needed.

Objectives / Topics Covered	• Model story problems within 100 • Connect a digit's place in a number to its value • Compare three-digit numbers using events • Build an analog clock using sequences and events
Lessons	Story Problems: Add and Subtract within 100 • Build a program that models a math story problem. Place Value: Ones, Tens, and Hundreds • Use events to connect a digit's place in a number to its value. Greater Than and Less Than: Three-Digit Numbers • Program a comparison of two three-digit numbers using events. Algebraic Thinking: Finding a Two-Digit Unknown • Build an interactive program that solves for a large unknown number in a number story. 3D Shapes • Draw 3D shapes and build a quiz game reviewing their attributes. Telling Time • Use sequences and events to create an analog clock showing time in digital and analog forms.

Thinking Across Disciplines - Science (Optional)

Optional lessons that use programming to explore second grade science concepts, including designs drawn from nature, changing landforms, moon phases, day and night, and seed dispersal. These can be taught in any order and integrated throughout the year as needed.

Objectives / Topics Covered	• Model how humans mimic plants and animals to solve problems • Sequence how events change Earth's surface • Use message events to model the phases of the Moon • Use loops to model the pattern of day and night
-----------------------------------	--

Lessons	Designing Solutions from Nature • Use events to show how humans survive in their environments by mimicking plants and animals. Changing Landforms • Sequence how a volcanic eruption changes Earth's surface. Phases of the Moon • Use message events to model the phases of the Moon. Sun and Moon, Day and Night • Use loops to model the movements of the sun and moon and the pattern of day and night. Seed Dispersal • Use message events and loops to model how an animal helps disperse seeds.
---------	--

Louisiana Computer Science: 2nd 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 .	