2nd Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The 2nd Grade Computer Science Course 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 36 lessons, each approximately 45 minutes long, providing a full year of instruction when taught once per week. While the course allows for instructional flexibility, some lessons are required to fully meet CSTA computer science standards and are clearly identified within the syllabus.

Standards Alignment Note: Lessons that list standards in parentheses next to the title are required to fully meet CSTA computer science standards. 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.

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/28414/overview

A clickable PDF can be found at https://codehs.com/CSRoadmaps

Course Breakdown

Unit 1: Optional Review (6 lessons)

These optional lessons are available for teachers to use as review or warm-up activities at the start of 2nd grade. They are not counted toward the 36-lesson course total and can be used selectively based on student readiness.

Objectives / Topics Covered	● Review foundational platform navigation and programming skills. ● Revisit events, loops, and message events from 1st grade. ● Build confidence before beginning new 2nd-grade content.
Lessons	**Welcome to CodeHop!** ● Learn how to log in and use the CodeHop Playground. This short introductory lesson can be used on its own or right before a full lesson. **Introduction to Programming** ● Explore the programming interface and add characters. **Events** ● Explain what an event is in programming and use multiple event blocks in a program. **Introduction to Repeat Loops** ● Use repeat loops to run a section of code multiple times. **Forever Loop Dance Party** ● Create sequences that are repeated while the program runs. **Introduction to Message Events** ● Program a relay race that uses messages to cause characters to interact.

Unit 2: Getting Started (1 lesson)

In this module, students apply computational thinking to analyze and decompose a real-world routine. This unplugged activity bridges prior knowledge and prepares students for the algorithmic thinking required throughout the 2nd-grade course.

Objectives / Topics Covered	● Apply computational thinking to decompose a routine into a step-by-step algorithm. ● Identify patterns and sequences in everyday activities.
Lessons	**Computational Thinking: School Day Routines (Unplugged)** (E2-ALG-PS-01) ● Use computational thinking concepts to identify patterns, break down tasks, sequence steps, and simplify processes in their school day routines.

Unit 3: Sequences & Events (3 lessons)

In this module, students build on prior sequencing and event skills in new contexts. Students navigate a grid, build a complete game, and practice debugging — developing precision, logic, and problem-solving skills.

Objectives / Topics Covered	● Use directional commands to navigate a character on a grid. ● Apply events and sequences to create an interactive game. ● Identify and fix bugs in programs using debugging strategies.
Lessons	**Introduction to the Grid** ● Use the grid feature to move characters to a specific location on the stage. **Tap-a-Mole Game** (2 classes) ● Create an interactive game using events. **Debugging: Events and Sequences** (E2-PRO-TR-07) ● Find and fix errors in provided code.

Unit 4: Loops (4 lessons)

In this module, students deepen their understanding of loops by applying them in new programming contexts. Students use loops to navigate paths, build timers, create animations with feedback, and design algorithms for grid-based challenges.

Objectives / Topics Covered	● Use loops to program efficient, repeated sequences of actions. ● Apply loops to create timing mechanisms and animations. ● Incorporate user feedback into loop-based programs. ● Design and code algorithms using loops to solve grid-based challenges.
Lessons	**Loops: Follow the Path** (E2-ALG-PS-01, E2-PRO-PD-04) ● Identify patterns and create a program using loops. **Making a Timer** ● Use loops, wait blocks, and turn blocks to create and compare two timers with different speeds. **Two-Step Dance & Feedback** (E2-PRO-RD-06) ● Create a program and revise it based on peer feedback and give attribution to a peer who helped improve their work. **Algorithms: Connecting a Path** (E2-ALG-PS-01) ● Create and adjust simple algorithms to move characters based on their size, shape, and starting position.

Unit 5: Message Events (5 lessons)

In this module, students expand their use of message events to coordinate complex interactions between characters and pages. Students build multi-part programs, practice debugging, and apply the full design process to create a polished project.

Objectives / Topics Covered	● Use message blocks to coordinate actions between multiple characters. ● Program repeating cycles using message events. ● Combine pages and message events to build multi-part programs. ● Debug programs that use message events and loops. ● Apply the design process to plan and create a programming project.
Lessons	**Message Events: Scout Plays in the Forest** ● Use message events to control the flow of a program. **Programming a Cycle** (E2-PRO-PD-04) ● Use message events to model a cycle. **Pages: Scout's Travels** ● Use messages to help Scout travel between pages in a program. **Debugging: Message Events and Loops** (E2-PRO-TR-07) ● Find and fix (debug) message event and loop errors in the provided code. **Exploring the Design Process** (E2-ALG-PS-01) ● Use the design process to plan, create, and improve a program with loops that models a solution to a simple real-world problem.

Unit 6: Culmination Projects (3 lessons)

In this module, students synthesize all programming skills learned throughout the year in a series of culminating game-design projects. Each project increases in complexity, guiding students through the full design process from concept to finished program.

Objectives / Topics Covered	● Review and apply a broad range of programming concepts in one project. ● Design and build a complete game using sequencing, events, and loops. ● Apply the design process to create a polished, multi-session game project.

Lessons	**Code Block Review** (2 classes | E2-PRO-RD-06) ● Use a variety of coding blocks in a program and explain their function within the program. **Racing Game** (2 classes | E2-PRO-PD-04) ● Create an interactive racing game with events, loops, and messages. **Moving Targets Game** (3 classes | E2-ALG-IM-03) ● Create a moving target game using sequences, events, and pages.

Unit 7: Computing Systems (2 lessons)

In this module, students explore how computer components connect and work together, and how computers communicate across networks. Students build a deeper understanding of hardware relationships and the infrastructure behind the Internet.

Objectives / Topics Covered	● Identify and describe how computer hardware components connect and interact. ● Explain how computers communicate through networks and the Internet.
Lessons	**Computer Basics: Connections (Unplugged)** (E2-SYS-HW-11) ● Learn what a computer is, how we use it, and what to do when it doesn't work. They will also be able to identify input, output, hardware, and software, and explain how they work together. **Exploring Computer Networks** ● Describe how networks connect devices to share information and model the sending and receiving of information using message blocks.

Unit 8: Safety & Responsibility (2 lessons)

In this module, students practice safe online behavior and responsible technology use. Through unplugged activities and real-world scenarios, students develop the digital citizenship skills needed to navigate technology safely and ethically.

Objectives / Topics Covered	● Practice safe online behavior including protecting personal information. ● Demonstrate responsible and respectful technology use in digital environments.
Lessons	**Responsible Digital Citizens (Unplugged)** (E2-SYS-SE-12) ● Explain what it means to be a responsible digital citizen, including understanding digital footprints, discussing cyberbullying, and knowing how to report concerns. **Practicing Responsible Technology Use (Unplugged)** (E2-SYS-IM-13) ● Demonstrate ways to use technology safely and responsibly.

Unit 9: Data & Analysis (3 lessons)

In this module, students collect, organize, analyze, and present data. Students explore how files and data are stored digitally, and conduct independent research to create data-driven programs that communicate their findings.

Objectives / Topics Covered	● Collect, organize, and analyze data to draw conclusions and identify patterns. ● Understand how digital files are organized and stored. ● Conduct independent research and present findings through a program.
Lessons	**Data Explorers** (E2-DAT-DC-08, E2-DAT-DI-09, E2-DAT-IM-10) ● Develop an investigative question and collect data using a survey. Then, students will create a program to present their collected data visually. **Managing Data Storage and Files**

	● Develop an investigative question and collect data using a survey. Then, students will create a program to present their collected data visually. **Choice Research** (2 classes | E2-DAT-IM-10) ● Collect and assess sources to answer a research question and communicate their findings visually.

Unit 10: Impacts of Computing (2 lessons)

In this module, students explore how computing shapes industries, careers, and society. Students discover real-world applications of computer science and examine how technology has transformed the world over time.

Objectives / Topics Covered	● Describe how computer science is used across different industries and careers. ● Explain how computing has changed society and examine its broad impacts.
Lessons	**Careers in CS: Coding for Fashion-Retail** (E2-SOC-CE-16) ● Explain how coding helps create and improve fashion designs and will create a program to design and animate a fashion character. **Impacts of Computing: Changing World** (E2-SOC-HI-14, E2-SOC-HU-15) ● Identify examples of computing devices in their daily lives and use programming to explain how technology has changed the way people live, work, and play.

Unit 11: AI Exploration (2 lessons)

In this module, students explore how artificial intelligence is used in everyday life and experience machine learning firsthand. Students discover how AI systems learn from examples to recognize patterns and make predictions.

Objectives / Topics Covered	● Identify ways AI is used to help people in everyday life. ● Explain how machine learning systems learn from data to recognize patterns.
Lessons	**How AI Helps Us (Unplugged)** (E2-SOC-HU-15) ● Explain how AI assistants answer questions by observing a demonstration and describe how AI helps people by solving problems and changing jobs. **Machine Learning: AutoDraw** (E2-ALG-ML-02) ● Describe how AutoDraw uses AI and a classifier to recognize and suggest drawings.

Unit 12: Preparing for Next Year (3 lessons)

In this module, students transition to the full CodeHop Blocks environment and are introduced to variables, a key concept in 3rd-grade programming. These lessons build a bridge between 2nd and 3rd-grade computer science skills.

Objectives / Topics Covered	● Navigate and explore the expanded CodeHop Blocks programming environment. ● Apply programming skills to create projects in the full block interface. ● Understand variables as a way to store and update information in a program.
Lessons	**From CodeHopJr to CodeHop Blocks** ● Navigate the basic interface of the CodeHop editor to create a simple program. **From CodeHopJr to CodeHop Blocks: Part 2** (E2-PRO-PD-04) ● Create a CodeHop program that uses an event and a loop. **Variables: Keeping Score** (E2-PRO-VD-05) ● Create a program that simulates keeping score using a variable.

2nd Grade Course 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**.	