Oklahoma Computer Science: 2nd Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Oklahoma Computer Science: 2nd Grade 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 state computer science standards and are clearly identified within the syllabus. All Digital Literacy lessons are required to ensure full standards alignment, as they address essential non-programming computer science concepts. Required lessons are labeled with the specific standards they address to support planning and compliance.

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 four-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/OK_2/overview



A clickable PDF can be found at https://codehs.com/OK-K-5Roadmaps

Course Breakdown

Optional Review

Students review core programming concepts from earlier grades — events, loops, and message events — while getting comfortable in the CodeHopJr environment.

Objectives / Topics Covered	● Navigate the CodeHopJr programming interface ● Create programs using events and sequences ● Use repeat and forever loops ● Program characters to interact with messages
Lessons	**Welcome to CodeHop!** ● Learn to log in and explore the CodeHopJr Playground. **Introduction to Programming** ● Navigate the programming interface to create a scene with 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 a dance sequence using a “repeat forever” loop. **Introduction to Message Events** ● Program a relay race that uses messages to make characters interact.

Unit 1: Getting Started (3 lessons)

Students learn how computer parts work together, practice safe and responsible technology use, and apply computational thinking to school day routines.

Objectives / Topics Covered	● Explain how input, output, hardware, and software work together ● Demonstrate safe and responsible technology use ● Apply computational thinking to everyday routines
Lessons	**Computer Basics: Connections (2.CS.D.01, 2.CS.HS.01, 2.CS.T.01, 2.IC.C.01)** ● Learn what a computer is and explain how input, output, hardware, and software work together. **Practicing Responsible Technology Use (2.IC.SI.01)** ● Demonstrate ways to use technology safely and responsibly. **Computational Thinking: School Day Routines** ● Break down school day routines into patterns and step-by-step sequences. **Getting Started Quiz (4 questions)**

Unit 2: Sequences and Events (7 lessons)

Students deepen their sequencing skills with the grid, algorithms, and debugging, and explore how data is stored, visualized, and used to make predictions.

Objectives / Topics Covered	● Find and fix errors in code ● Use the grid to move characters precisely ● Create and adjust simple algorithms ● Identify patterns and trends in data
Lessons	**Debugging: Events and Sequences (2.AP.PD.03, 2.AP.PD.04)** ● Find and fix errors in provided code.

	Introduction to the Grid (2.AP.A.01, 2.AP.C.01) ● Use the grid feature to move characters to specific locations on the stage. **Managing Data Storage and Files (2.DA.S.01)** ● Recognize that computers store data as files and model how data is collected and stored. **Algorithms: Connecting a Path** ● Create and adjust simple algorithms to move characters based on their size, shape, and starting position. **Data Patterns and Predictions (2.DA.IM.01)** ● Identify patterns and trends in data visualizations and create a program to communicate predictions. **Tap-a-Mole Game (2 classes | 2.AP.C.01, 2.AP.PD.03, 2.AP.PD.04)** ● Create an interactive tap-a-mole game using events. **Sequences and Events Quiz (4 questions)**

Unit 3: Message Events (5 lessons)

Students use message events to control program flow, model cycles and networks, travel between pages, and keep score with variables.

Objectives / Topics Covered	● Use message events to control the flow of a program ● Model cycles and real-world processes ● Keep score using variables ● Describe how networks share information
Lessons	**Message Events: Scout Plays in the Forest (2.AP.A.01, 2.AP.C.01)** ● Use message events to control the flow of a program. **Programming a Cycle (2.AP.A.01, 2.AP.C.01, 2.AP.PD.01)** ● Use message events to model a cycle. **Pages: Scout's Travels (2.AP.A.01, 2.AP.C.01)** ● Use messages to help Scout travel between pages in a program. **Variables: Keeping Score (2.AP.V.01)** ● Create a program that simulates keeping score using a variable. **Exploring Computer Networks (2.NI.NCO.01)** ● Describe how networks connect devices and model sending and receiving information with message blocks. **Message Events Quiz (4 questions)**

Unit 4: Loops (6 lessons)

Students combine loops with patterns, timers, and animations, practice debugging loop errors, and use the design process to solve a real-world problem.

Objectives / Topics Covered	● Identify patterns and program them with loops ● Debug message event and loop errors ● Use the design process to plan and improve programs
Lessons	**Loops: Follow the Path (2.AP.C.01)** ● Identify patterns and create a program using loops. **Debugging: Message Events and Loops (2.AP.PD.03, 2.AP.PD.04)** ● Find and fix message event and loop errors in provided code. **Making a Timer (2.AP.C.01, 2.AP.PD.03)** ● Use loops, wait blocks, and turn blocks to create and compare two timers with different speeds. **Original Story Animations - Fiction Story (2 classes | 2.AP.A.01, 2.AP.C.01, 2.AP.PD.01)** ● Develop an original story and create a program to animate it.

	Exploring the Design Process ● Use the design process to plan, create, and improve a program that models a solution to a real-world problem. **Loops Quiz (4 questions)**

Unit 5: Culmination Projects (10 lessons)

Students apply everything they have learned to build games and projects that combine events, loops, messages, pages, and more.

Objectives / Topics Covered	● Combine a variety of coding blocks in one program ● Build interactive games with events, loops, and messages ● Design and share a multi-page game
Lessons	**Code Block Review (2 classes | 2.AP.A.01, 2.AP.C.01, 2.AP.PD.03)** ● Use a variety of coding blocks in a program and explain their function. **Racing Game (2 classes | 2.AP.A.01)** ● Create an interactive racing game with events, loops, and messages. **Moving Targets Game (3 classes | 2.AP.C.01, 2.AP.PD.03, 2.AP.PD.04)** ● Create a moving-target game using sequences, events, and pages. **Design an Adventure Game (3 classes | 2.AP.A.01, 2.AP.C.01, 2.AP.PD.01, 2.AP.PD.03)** ● Create a story-based, multi-page adventure game using the skills they have learned.

Unit 6: Digital Literacy (5 lessons)

Students learn to protect login information, collect and present survey data, and research a question using trustworthy sources.

Objectives / Topics Covered	● Keep usernames and passwords safe ● Collect data with surveys and present it visually ● Assess sources to answer a research question
Lessons	**Password Protectors (2.NI.C.01)** ● Demonstrate strategies to keep usernames and passwords safe. **Data Explorers (2 classes | 2.DA.CVT.01, 2.DA.IM.01)** ● Develop an investigative question, collect survey data, and present it visually. **Choice Research (2 classes | 2.AP.A.01, 2.AP.C.01, 2.AP.PD.02)** ● Collect and assess sources to answer a research question and communicate findings visually. **Digital Literacy Quiz (4 questions)**

Unit 7: Preparing for Next Year

Students preview next year's concepts with unplugged conditionals activities and take their first steps from CodeHopJr into the CodeHop block editor.

Objectives / Topics Covered	● Build instruction sequences with conditionals ● Navigate the CodeHop block editor ● Create CodeHop programs with events and loops
Lessons	**Coding Card Game: Conditionals** ● Work together to build a sequence of instructions with conditionals that moves Scout through a maze. **Coding Card Game: Conditionals 2** ● Practice building longer maze sequences with conditionals in the coding card game. **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 ● Create a CodeHop program that uses an event and a loop.

Oklahoma 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**.	