



Oklahoma Computer Science: 4th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Oklahoma Computer Science: 4th 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.

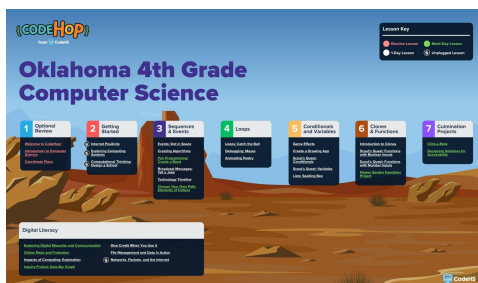
The course includes 36 lessons, each approximately 45 minutes long, providing a full year of instruction when taught once per week. While the course allows for instructional flexibility, some lessons are required to fully meet state computer science standards and are clearly identified within the syllabus. 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 five-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_4/overview



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

Course Breakdown

Optional Review

Students get comfortable in the CodeHop editor by reviewing computer science vocabulary, building a simple program, and animating with the coordinate plane.

Objectives / Topics Covered	• Navigate the CodeHop Playground • Define important computer science vocabulary • Animate using the coordinate plane
Lessons	Welcome to CodeHop! • Learn to log in and explore the CodeHop Playground. Introduction to Computer Science • Define important computer science vocabulary and create a simple program. The Coordinate Plane • Create an open-ended animation using the coordinate plane.

Unit 1: Getting Started (3 lessons)

Students learn how to spread positivity online, explore the parts of computing systems, and apply computational thinking to a design challenge.

Objectives / Topics Covered	• Create a code of conduct for responsible online behavior • Identify parts of a computing system • Identify simple hardware and software problems • Apply computational thinking to a design task
Lessons	Internet Positivity (4.IC.SI.01, 4.NI.C.01) • Explain how actions can spread positivity online and create a code of conduct for responsible behavior. Exploring Computing Systems (4.CS.HS.01, 4.CS.T.01) • Identify parts of the computing system and identify simple hardware and software problems. Computational Thinking: Design a School • Use computational thinking to design a school. Getting Started Quiz (5 questions)

Unit 2: Sequences & Events (8 lessons)

Students program with events, algorithms, and broadcast messages, collaborate through pair programming, and explore how technology shapes culture.

Objectives / Topics Covered	• Create programs with multiple types of events • Program and compare multiple algorithms • Collaborate through pair programming • Explore how technology influences culture
Lessons	Events: Dot in Space (4.AP.C.01, 4.AP.M.01, 4.AP.PD.03) • Create and explain a program using multiple types of event blocks. Creating Algorithms (4.AP.A.01, 4.AP.C.01) • Program multiple algorithms and assess which one best meets their needs. Pair Programming: Create a Band (2 classes 4.AP.C.01, 4.AP.M.01, 4.AP.PD.01, 4.AP.PD.03, 4.IC.SI.02) • Collaborate through pair programming to design and code a band using keyboard inputs.

	Broadcast Messages: Tell a Joke (4.AP.C.01, 4.AP.M.01, 4.AP.PD.03) • Use broadcast messages to program two sprites to tell a knock-knock joke. Technology Timeline (2 classes 4.IC.C.01) • Create an interactive timeline of music player technology and explain its influence on cultural practices. Choose Your Own Path: Elements of Culture (2 classes 4.AP.C.01, 4.AP.M.01, 4.AP.M.02, 4.AP.PD.02, 4.AP.PD.03) • Identify elements of culture while creating a cultural choose-your-own-path game. Sequences and Events Quiz (5 questions)
--	---

Unit 3: Loops (3 lessons)

Students use different types of loops to build games and animations and practice decomposing programs to debug them.

Objectives / Topics Covered	• Use two types of loops in a game • Decompose a program to debug it • Animate a poem with events and loops
Lessons	Loops: Catch the Ball (4.AP.C.01, 4.AP.PD.03) • Use two types of loops to create a simple game. Debugging: Mazes • Decompose a program to debug it and make it run as intended. Animating Poetry • Create an animated reading of a poem using events and loops. Loops Quiz (5 questions)

Unit 4: Conditionals & Variables (5 lessons)

Students use conditionals, variables, and lists to build interactive programs — from drawing apps to games — and refine them with peer feedback.

Objectives / Topics Covered	• Create programs using if/then conditionals • Use variables to track points • Program keyboard and mouse inputs • Use lists in a game
Lessons	Game Effects (4.AP.C.01, 4.AP.M.02) • Modify a game to add engaging effects and update it based on peer feedback. Create a Drawing App (4.AP.C.01) • Create a drawing app by programming keyboard and mouse inputs, loops, and conditional statements. Scout's Quest: Conditionals (4.AP.C.01, 4.AP.PD.03) • Create a program using if/then conditionals. Scout's Quest: Variables • Create and use variables to track points in a program. Lists: Spelling Bee • Use lists to create a spelling bee game. Conditionals and Variables Quiz (5 questions)

Unit 5: Clones & Functions (5 lessons)

Students create animations with clones and learn to define and call functions with boolean and number inputs.

Objectives / Topics Covered	• Create animations using clones • Create functions with boolean and number inputs • Use functions to draw repeated designs
-----------------------------------	---

Lessons	Introduction to Clones (4.AP.C.01, 4.AP.V.01) - Create an animation using clones and investigate the limitations of the program. Scout's Quest: Functions with Boolean Inputs (4.AP.C.01, 4.AP.PD.03) - Create a function with a boolean input that acts on whether a password is correct. Scout's Quest: Functions with Number Inputs (4.AP.C.01, 4.AP.M.01, 4.AP.PD.03) - Create a drawing using functions with number inputs. Flower Garden Functions Project (2 classes 4.AP.C.01, 4.AP.M.01, 4.AP.PD.03) - Create a function in a program to draw multiple flowers. Clones and Functions Quiz (5 questions)
---------	---

Unit 6: Culmination Projects (4 lessons)

Students apply conditionals, variables, booleans, and events in a full game build, then redesign a game for accessibility using design thinking.

Objectives / Topics Covered	- Build an interactive game with conditionals, variables, and events - Use the design thinking process - Improve accessibility and usability for diverse users
Lessons	Click-a-Mole (2 classes) - Create an interactive Whack-a-Mole style game using conditionals, variables, booleans, and events. Designing Solutions for Accessibility (2 classes 4.AP.C.01, 4.AP.M.01, 4.AP.PD.01, 4.AP.PD.03, 4.AP.PD.04, 4.AP.V.01, 4.IC.C.02) - Redesign a game to improve accessibility and usability for diverse users using design thinking.

Unit 7: Digital Literacy (8 lessons)

Students practice digital etiquette and online safety, investigate technology's impacts, run a data inquiry, and learn how files, attribution, and networks work.

Objectives / Topics Covered	- Communicate with proper digital etiquette - Practice secure online habits - Display investigation results with a program - Model how information travels across networks
Lessons	Exploring Digital Etiquette and Communication (4.IC.SI.01) - Demonstrate proper digital etiquette when communicating in an online community. Online Risks & Protection - Practice secure habits and learn about the tools and technologies that protect information online. Impacts of Computing: Exploration (4.IC.C.01) - Create a project comparing past and present technology, identifying a trend, and explaining positive and negative impacts. Inquiry Project: Data Bar Graph (2 classes 4.AP.V.01, 4.DA.CVT.01, 4.DA.IM.01) - Follow the inquiry process and modify a program to display the results of an investigation. Give Credit When You Use It (4.IC.SLE.01) - Search for information to answer questions online and provide proper attribution to sources. File Management and Data in Action (4.DA.S.01) - Explain how different types of digital data take up different amounts of space and can be stored in different ways. Networks, Packets, and the Internet (4.CS.HS.01, 4.NI.NCO.01) - Compare wired and wireless network connections and model how devices transfer information using rules and packets. Digital Literacy Quiz (5 questions)

Oklahoma 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 .	