



Oklahoma Computer Science: 3rd Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Oklahoma Computer Science: 3rd 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.

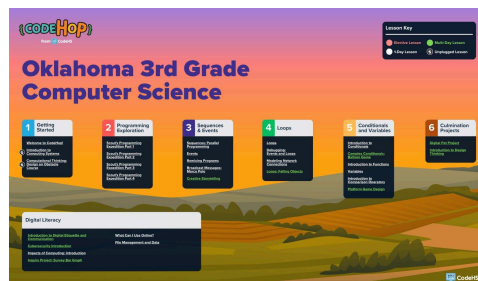
Learning Environment: This course is teacher-led and includes ready-to-use lessons following a consistent structure: Introduction, Guided Practice, Independent Practice, Extension, and Reflection. Instruction follows an “I do, we do, you do” model and incorporates spiral review to reinforce concepts 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. Most 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.

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_3/overview



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

Course Breakdown

Unit 1: Getting Started (2 lessons)

Students log in to CodeHop, learn the parts of a computing system, and use computational thinking to design solutions.

Objectives / Topics Covered	- Identify parts of a computing system - Solve simple hardware and software problems - Apply computational thinking to a design task
Lessons	Welcome to CodeHop! - Learn to log in and explore the CodeHop Playground. Introduction to Computing Systems (3.CS.HS.01, 3.CS.T.01, 3.IC.C.02) - Identify parts of the computing system and solve simple hardware and software problems. Computational Thinking: Design an Obstacle Course (3.AP.M.01) - Use computational thinking to design an obstacle course.

Unit 2: Programming Exploration (4 lessons)

Through Scout's story-driven programming expedition, students learn to program sprites with motion, looks, events, and loops in the CodeHop editor.

Objectives / Topics Covered	- Program sprites to move and talk - Build sequences to animate stories - Combine loops, events, looks, and motion blocks
Lessons	Scout's Programming Expedition: Part 1 (3.AP.M.01, 3.AP.PD.03) - Use basic programming commands to program a sprite to move and talk. Scout's Programming Expedition Part 2 (3.AP.M.01, 3.AP.PD.03) - Add sprites and create a sequence to animate a story. Scout's Programming Expedition Part 3 (3.AP.M.01, 3.AP.PD.03) - Create an animated Scout story using loops, events, looks, and motion blocks. Scout's Programming Expedition Part 4 - Create an animated Scout story using events, looks, and motion blocks.

Unit 3: Sequences & Events (6 lessons)

Students build programs with parallel sequences, events, and broadcast messages, learn to remix programs with proper credit, and animate original stories.

Objectives / Topics Covered	- Create programs with parallel sequences - Trigger actions using events and broadcast messages - Remix programs and credit original creators - Plan and animate a story
Lessons	Sequences: Parallel Programming (3.AP.M.01, 3.AP.PD.03) - Create a program using parallel sequences. Events (3.AP.C.01) - Create a program using events. Remixing Programs (3.AP.M.02, 3.AP.PD.02, 3.IC.SI.01, 3.IC.SLE.01, 3.NI.C.01) - Remix digital programs and give appropriate credit to original creators. Broadcast Messages: Marco Polo (3.AP.C.01, 3.AP.M.01, 3.AP.PD.03) - Use broadcast messages to trigger actions between sprites in a program.

	Creative Storytelling (2 classes 3.AP.C.01, 3.AP.M.01) Plan and animate a story using events and sequences.
--	--

Unit 4: Loops (5 lessons)

Students use and compare different kinds of loops, debug programs with events and loops, and model how data travels across networks.

Objectives / Topics Covered	- Use loops to repeat instructions - Decompose programs to debug them - Compare the advantages of different loop algorithms - Model how messages move between devices
Lessons	Loops (3.AP.C.01, 3.AP.PD.03) - Explain that a loop repeats one or more instructions and use loops in CodeHop. Debugging: Events and Loops (3.AP.PD.03) - Decompose a program to debug it and make it run as intended. Modeling Network Connections (3.NI.NCO.01) - Describe how messages move between devices and create a program showing how data travels. Loops: Falling Objects (2 classes 3.AP.A.01, 3.AP.M.02, 3.AP.PD.03) - Create a program using different loops and compare the advantages of each loop algorithm.

Unit 5: Conditionals and Variables (7 lessons)

Students learn to control programs with conditionals, comparison operators, functions, and variables, then apply them to build a platform game.

Objectives / Topics Covered	- Create programs with if/then and if/then/else blocks - Create and use functions - Create and change the value of variables - Use comparison operators in conditions
Lessons	Introduction to Conditionals (3.AP.C.01) - Explain what a conditional is and create a program that uses if/then blocks. Complex Conditionals: Balloon Game (3.AP.C.01) - Use “if/then/else” conditionals in a program. Introduction to Functions (3.AP.C.01, 3.AP.PD.03) - Create and use functions in a program. Variables (3.A.V.01, 3.AP.C.01) - Explain what a variable is and create and change its value in a program. Introduction to Comparison Operators (3.A.V.01, 3.AP.C.01) - Use comparison operators as the condition in an if/else block. Platform Game Design (2 classes 3.AP.C.01, 3.AP.PD.01, 3.AP.PD.03, 3.AP.PD.04, 3.IC.SI.02) - Build a platform game with keyboard events and a forever loop, then debug and improve it.

Unit 6: Culmination Project (5 lessons)

Students bring their skills together in a digital pet project and use design thinking to make tools accessible to more users.

Objectives / Topics Covered	- Combine events, conditionals, variables, and broadcasts in one project - Use the design thinking process - Make a tool accessible to more users
-----------------------------	---

Lessons	Digital Pet Project (3 classes 3.A.V.01, 3.AP.C.01, 3.AP.M.01, 3.AP.M.02, 3.AP.PD.01, 3.AP.PD.03, 3.IC.SI.02) • Create a digital pet using events, conditionals, variables, comparison operators, and broadcasts. Introduction to Design Thinking (2 classes 3.A.V.01, 3.AP.C.01, 3.AP.M.01, 3.AP.PD.01, 3.AP.PD.03, 3.AP.PD.04, 3.IC.C.02) • Use the design thinking process to make a tool accessible to more users.
---------	--

Unit 7: Digital Literacy (7 lessons)

Students practice digital etiquette and online safety, explore how technology and culture influence each other, run a survey inquiry, and learn how digital data is stored and attributed.

Objectives / Topics Covered	• Communicate with proper digital etiquette • Recognize online threats and stay safe online • Present survey results with a program • Search online and give proper attribution
Lessons	Introduction to Digital Etiquette and Communication (3.IC.SI.01) • Demonstrate proper digital etiquette when communicating in an online community. Cybersecurity Introduction (3.NI.C.01) • Recognize online threats and identify how to stay safe online. Impacts of Computing: Introduction (3.IC.C.01) • Explain how technology and culture influence each other through a past-and-present technology project. Inquiry Project: Survey Bar Graph (2 classes 3.A.V.01, 3.DA.CVT.01, 3.DA.IM.01) • Follow the inquiry process and modify a program to display the results of an investigation. What Can I Use Online? • Search for information to answer questions online and provide proper attribution to sources. File Management and Data (3.DA.S.01) • Explain how different types of digital data take up space and identify where data can be stored.

Oklahoma Computer Science: 3rd 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 .	