Oklahoma Computer Science: 5th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Oklahoma Computer Science: 5th 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_5/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, the coordinate plane, and drawing tools.

Objectives / Topics Covered	● Navigate the CodeHop Playground ● Define important computer science vocabulary ● Animate using the coordinate plane ● Create custom sprites and backdrops
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. **Drawing Tools** ● Create customized sprites and backdrops using the drawing tools.

Unit 1: Getting Started (3 lessons)

Students plan for balanced, responsible technology use, troubleshoot computing systems, and apply computational thinking to a design challenge.

Objectives / Topics Covered	● Create a personal screen time plan ● Identify parts of a computing system ● Identify simple hardware and software problems ● Apply computational thinking to a design task
Lessons	**Safe and Responsible Use of Technology (5.IC.SI.01)** ● Explain the importance of balancing screen time by creating a personal screen time plan. **Practicing with Computing Systems (5.CS.HS.01, 5.CS.T.01)** ● Identify parts of the computing system and identify simple hardware and software problems. **Computational Thinking: Design a Neighborhood (5.AP.M.01)** ● Use computational thinking to design a neighborhood. **Getting Started Quiz (5 questions)**

Unit 2: Sequence & Events (3 lessons)

Students program with events, compare and refine algorithms for efficiency, and decompose programs to debug them.

Objectives / Topics Covered	● Use events in a program ● Compare and refine algorithms for the same task ● Decompose a program to debug it
Lessons	**Events: Traveling with Scout (5.AP.C.01, 5.AP.M.01, 5.AP.PD.03)** ● Use events in a program. **Compare and Refine Algorithms (5.AP.A.01, 5.AP.C.01)** ● Compare and refine multiple algorithms for the same task to determine the most efficient. **Debugging: Make a Pizza (5.AP.M.01, 5.AP.PD.03)** ● Decompose a program to debug it and make it run as intended.

	Sequences and Events Quiz (5 questions)

Unit 3: Loops (3 lessons)

Students use loops with the pen tool to draw turtle graphics and program multi-scene animations with repeat loops.

Objectives / Topics Covered	● Create looping turtle graphics with the pen tool ● Use repeat loops to program an animation ● Build an animation with multiple scenes
Lessons	**Creating Turtle Graphics (5.AP.C.01, 5.AP.M.01, 5.AP.PD.03)** ● Use the pen tool to create looping turtle graphics. **Animation Loops Project (2 classes | 5.AP.M.02, 5.AP.PD.03)** ● Use repeat loop blocks to program an animation with multiple scenes. **Loops Quiz (5 questions)**

Unit 4: Conditionals & Variables (7 lessons)

Students plan programs with pseudocode and decomposition, control programs with conditionals and variables, build with lists and operators, and practice proper attribution.

Objectives / Topics Covered	● Create programs that use conditionals ● Plan programs with pseudocode and decomposition ● Use variables, lists, and operators ● Give appropriate attribution when remixing programs
Lessons	**Conditionals: Mazes (5.AP.C.01)** ● Create a program that uses conditionals and plan it with pseudocode. **Plan a Quest (2 classes | 5.AP.C.01, 5.AP.M.01, 5.AP.M.02, 5.AP.PD.01, 5.AP.PD.03)** ● Plan and decompose the steps needed to create a quest program. **Variables in Dance (5.AP.C.01, 5.AP.PD.03, 5.AP.V.01)** ● Use variables to control pitch and dance speeds in a program. **Giving Credit Through Attributions (5.AP.M.02, 5.AP.PD.02, 5.IC.SI.01, 5.IC.SLE.01)** ● Give appropriate attribution when creating or remixing programs online. **Lists: Shopping with Scout (2 classes)** ● Create a shopping simulator using variables, lists, and operators. **Conditionals and Variables Quiz (5 questions)**

Unit 5: Clones & Functions (6 lessons)

Students build games with clones, classes, and objects, and create functions with inputs to organize their programs.

Objectives / Topics Covered	● Create a game using clones ● Use classes, objects, and randomizers in games ● Create and use functions with inputs
Lessons	**Clones: Throwing Acorns Game** ● Create a throwing acorns game using clones. **Classes and Objects in Games (2 classes | 5.AP.C.01, 5.AP.M.01, 5.AP.M.02, 5.AP.PD.01, 5.AP.PD.03, 5.AP.PD.04, 5.AP.V.01)** ● Create an interactive game using classes, objects, and randomizers to change object characteristics. **Functions: About Me (5.AP.C.01, 5.AP.M.01, 5.AP.PD.03, 5.AP.V.01)** ● Create and use a function with input in a program.

	Functions: Paint by Numbers (2 classes | 5.AP.C.01, 5.AP.M.01, 5.AP.PD.03, 5.AP.V.01) ● Create and use a function that takes input in a program. **Clones and Functions Quiz (5 questions)**

Unit 6: Culmination Project (3 lessons)

Students use the design thinking process to design and prototype an app that solves a real user's need.

Objectives / Topics Covered	● Use the design thinking process ● Design an app that solves a user's need ● Test and share a prototype
Lessons	**Design an App (3 classes | 5.AP.C.01, 5.AP.M.01, 5.AP.PD.01, 5.AP.PD.03, 5.AP.PD.04, 5.AP.V.01, 5.IC.C.02)** ● Use the design thinking process to design an app that helps solve a user's need.

Unit 7: Digital Literacy (11 lessons)

Students explore cybersecurity laws, technology's cultural impacts, and digital etiquette, run data inquiries and research projects, and learn how files and networks organize and move information.

Objectives / Topics Covered	● Explain cybersecurity policies and laws ● Communicate with proper digital etiquette ● Display investigation results with a program ● Model how devices communicate across networks
Lessons	**Cybersecurity Policies and Laws (5.NI.C.01)** ● Explain school technology policies and research a state cybersecurity law. **Impacts of Computing: Innovation (5.IC.C.01)** ● Create a project comparing past and present technology and predicting how emerging technologies may affect communities. **Digital Etiquette and Teamwork (5.IC.SI.01)** ● Demonstrate proper digital etiquette when communicating in an online community. **Inquiry Project: Line Graph (2 classes | 5.AP.V.01, 5.DA.CVT.01, 5.DA.IM.01)** ● Follow the inquiry process and modify a program to display the results of an investigation. **Research: Informational Programs (2 classes | 5.DA.CVT.01, 5.DA.IM.01)** ● Examine information from different sources and create a PSA on healthy sleep habits. **File Management and Data Exploration (5.DA.S.01)** ● Explain how different types of digital data take up space and identify where data can be stored. **Collaborating Globally (2 classes | 5.IC.SI.02)** ● Collaborate with people from another community to create a digital tourism advertisement. **Network Connections (5.CS.HS.01, 5.NI.NCO.01)** ● Compare wired and wireless networks and model how devices transfer information using rules and packets. **Digital Literacy Quiz (5 questions)**

Oklahoma Computer Science: 5th 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**.	