



CodeHS

Oklahoma Computer Science 1st Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Oklahoma Computer Science 1st Grade Course** introduces students to foundational programming concepts through **ScratchJr**, 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 designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging projects to showcase student understanding.

The lessons are delivered in an **"I do, we do, you do"** format, ensuring a gradual release of responsibility and fostering confidence in students as they learn. Teachers can adapt the content to fit their schedule and instructional needs. The concepts taught in this course spiral across grade levels, ensuring that students can revisit and build upon their understanding year after year, even if all lessons are not completed within a single year. The course includes a total of 36 **contact hours**, each approximately 30-45 minutes long. This provides a full school year of material if teaching one lesson per week. Optional digital literacy lessons are also available to complement the programming curriculum with non-programming computer and technology skills.

Programming Environment: Students will write and run programs in **ScratchJr** embedded and 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_1/overview

Course Breakdown

Optional Review

This optional review unit is designed to support students who need more time exploring ScratchJr or who would benefit from additional practice before or after completing core lessons. The Scout Adventures lessons offer a sequential, story-based experience to reinforce key skills in a fun and engaging way.

Objectives / Topics Covered	• Log in and navigate the Playground. • Explore and interact with the ScratchJr environment. • Use motion blocks and sequencing to animate characters.
Lessons	Welcome to CodeHS! (15 minute Lesson) • Learn how to log in and explore the Playground; a brief introduction that can be used on its own or before starting a full lesson.

	Scout Adventures 1: Introducing Scout - Explore the ScratchJr interface and practice adding characters to the stage. Scout Adventures 2: Scout Starts Exploring - Add backgrounds and create a new page in ScratchJr to build out a story. Scout Adventures 3: Scout Meets a Friend - Delete and modify characters, building confidence in editing elements in a project. Scout Adventures 4: Scout Explores the Forest - Use motion blocks to move characters across the stage and begin creating animations. Scout Adventures 5: Scout and Bluebird Help - Build a sequence of motion blocks to guide characters through a task, such as collecting items. Scout Adventures 6: Scout Celebrates with Friends - Create a celebration scene by combining characters, pages, backgrounds, and events using motion blocks.
--	--

Unit 1: Getting Started (3 weeks)

In this unit, students will explore the basics of computing by learning the functions of computers and their components, practice identifying positive and negative online behaviors, and apply foundational computational thinking skills like pattern recognition, sequencing, and task decomposition to real-life routines.

Objectives / Topics Covered	- Recognize the basic parts of a computer. - Explore safe and responsible ways to use technology. - Practice computational thinking in everyday routines.
Lessons	Computer Basics: Exploration - Learn what a computer is, how we use it, what to do when it doesn't work, and identify parts like input, output, hardware, and software. Exploring Responsible Technology Use - Understand how to use technology safely and respectfully at school and home. Computational Thinking: Evening Routines - Practice thinking like a computer scientist by finding patterns, breaking down tasks, and sequencing steps using familiar routines.

Unit 2: Sequences and Events (9 weeks)

In this unit, students explore how to use sequences and events to control the behavior of characters. They will build animated scenes, games, and interactive projects while learning how to respond to events, pause actions, and debug errors.

Objectives / Topics Covered	- Use drawing tools in ScratchJr. - Create simple sequences. - Create programs using motion, hide, grow/shrink, and wait blocks. - Identify and fix errors in sequences of code.
Lessons	Drawing Tools: Nature Walk - Use ScratchJr's drawing tools to create a nature-themed scene. Events - Learn what an event is in programming and use event blocks to trigger actions in a program. Sequences: Digital Responsibilities - Create a program using sequences to show how to act responsibly online. Basic Data and Programming - Collect simple data and present it visually. Hide and Seek Game - Use the hide block to build an interactive game where characters disappear and reappear.

	Grow and Shrink Blocks in Motion - Combine motion with grow and shrink blocks to animate characters. Introduction to the Wait Block - Use the wait block to make characters pause before taking action in a program. Introduction to Debugging - Identify and fix mistakes in a sequence of code to make a program run correctly. Positive Online Behavior - Learn to recognize appropriate and inappropriate online behavior by exploring digital interactions.
--	--

Optional Unplugged Loops Activities

This optional unplugged unit gives students a hands-on way to practice programming concepts without using devices. By working together to move Scout through a maze using coding cards, students reinforce sequencing and looping skills in a fun, collaborative setting.

Objectives / Topics Covered	- Build sequences that include loops. - Understand how loops can simplify repeated actions in a program.
Lessons	Coding Card Game: Loops - Work together to build a sequence of instructions using loops to help a character move through a maze. Coding Card Game: Loops 2 - Continue practicing with loops by creating new sequences to solve different maze challenges as a team.

Unit 3: Loops (3 weeks)

In this unit, students will learn how to make characters repeat actions using loops. They'll use repeat and forever loops to build fun animations, games, and stories. By the end of the unit, students will be able to recognize when and how to use loops to make their code simpler and more efficient.

Objectives / Topics Covered	- Build sequences that include loops. - Use “forever loops” to create continuous action.
Lessons	Introduction to Repeat Loops - Use repeat loops to run a section of code multiple times and simplify repeated actions. Loops: Catching Butterflies - Create a butterfly-catching game using loops with “show” and “hide” blocks to repeat actions. Forever Loop Dance Party - Build a fun animation where characters repeat actions using the “repeat forever” loop.

Unit 4: Message Events (4 weeks)

In this unit, students will learn how to make characters talk to each other using message events. They'll explore how to send and receive messages to create interactions between characters, bring stories to life, and build fun, interactive projects.

Objectives / Topics Covered	- Use message events to make characters interact. - Use speed blocks to control movement.
-----------------------------------	--

Lessons	Introduction to Message Events - Program a relay race where characters use message blocks to interact with each other in a sequence. Message Events: Simon Says - Use message events to make one character send messages that trigger actions in multiple others. Create an Original Story Animation - Design and program an animated story using characters, messages, and sequences of actions. Speed Block: Bouncy Ball - Use speed blocks and message events to control how fast a character moves across the screen.
---------	---

Unit 5: Pages (3 weeks)

In this unit, students will learn how to use the “go to page” block in ScratchJr to create multi-page projects. They’ll design games and animations that move from one scene to another, building on their knowledge of events and loops while adding creativity through digital storytelling and design.

Objectives / Topics Covered	- Use “go to page” blocks to move between pages. - Combine message events and loops.
Lessons	Pages: Create a Tapping Game - Create a simple game that moves between pages using “go to page” blocks. Create a Mini Golf Game - Build a mini golf game by combining message events and loops to animate gameplay across pages. Digital Greeting Card - Design a digital greeting card using events and loops to bring the message to life with animations.

Unit 6: Grid (4 weeks)

In this unit, students will learn how to use the grid in ScratchJr to help plan and control character movement. They’ll design mazes, create animations, and explore how to end actions using the “end” block, all while practicing spatial reasoning and precise programming.

Objectives / Topics Covered	- Use the grid to program character movements. - Use the “end” block to signal when a sequence or event is finished.
Lessons	Grid: Solving Mazes - Design a maze and use the grid to program a character’s movement through it. End Block: Program a Race - Create an animated race and use the “end” block to signal when the race is finished. Grid: Arctic Animation (2 part lesson) - Use the grid to place and move characters precisely while creating an Arctic-themed animation.

Unit 7: Culmination Projects (5 weeks)

In this unit, students will apply everything they’ve learned throughout the course to create original projects that showcase their coding skills and creativity.

Objectives / Topics Covered	- Plan and create a personalized program. - Adjust levels of difficulty in a program.
-----------------------------	--

Lessons	About Me Project (3 part lesson) - Plan and create a program that shares personal characteristics, interests, and favorite things using ScratchJr. River Crossing Game (2 part lesson) - Build a game with moving obstacles and adjust the difficulty level using speed blocks and programming logic.
---------	---

Unit 8: Digital Literacy (5 weeks)

In this unit, students explore how to behave responsibly online, collect and present data through simple programs, and conduct guided research using trusted sources. They also learn how information is stored and shared, and model how people and devices communicate over networks and the internet.

Objectives / Topics Covered	- Identify appropriate and inappropriate online behavior. - Collect and organize simple data, then present it visually. - Conduct guided research and communicate findings visually. - Understand how networks and the internet allow people and devices to connect and share information.
Lessons	Guided Research (2 part lesson) - Use research tools to find information and create a program that communicates findings. Variables: Storing Data with Symbols - Model how a program stores data using variables. Data Storage and Files Practice - Understand how to store and manage files using digital devices. Networks and the Internet - Understand what a network is and model how people and devices communicate over networks and the Internet.

Oklahoma Computer Science 1st 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.
All of these resources and more are found on the Elementary Resources Page .	