

Virginia Computer Science 1st Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Virginia 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 lessons, with each lesson approximately 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 the CodeHop platform. 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/27180/overview



A clickable PDF can be found at https://codehs.com/VA-CSRoadmaps

Course Breakdown

Optional Review

This optional module supports students in revisiting key skills such as logging in, navigating ScratchJr, and reviewing basic programming concepts introduced in previous lessons.

Objectives / Topics Covered	 Refresh familiarity with the CodeHop Playground and login process. Revisit character, page, and background creation in ScratchJr. Reinforce sequences and motion block usage in a narrative structure.
Lessons	 Welcome to CodeHop! (15 minute lesson) ■ Learn how to log in and navigate the Playground. Can be used as a warm-up or review. Scout Adventures 1-6 ● Follow Scout the Squirrel through a six-part interactive story that helps students build foundational ScratchJr skills.

Unit 1: Getting Started (3 lessons)

Students are introduced to basic computer concepts, responsible technology use, and computational thinking practices. This unit establishes a foundation in safe technology use, computer literacy, and problem-solving skills that students will carry into later lessons.

Objectives / Topics Covered	 Identify the parts of a computer and explain how we use them. Describe safe and responsible ways to use technology. Apply computational thinking skills to everyday routines. 	
Lessons	Computer Basics: Exploration • Learn how computers work and understand hardware, software, input, and output. Exploring Responsible Technology Use • Understand how to use technology safely and responsibly. Computational Thinking: Evening Routines • Apply pattern recognition and sequencing to daily tasks.	

Unit 2: Sequences & Events (7 lessons)

Students explore how sequences and events form the foundation of programming. They will use ScratchJr's drawing tools, event blocks, and sequence-building to design interactive animations and games. This unit also introduces debugging strategies and highlights real-world connections to responsible technology use and careers in computer science.

Objectives / Topics Covered	 Use drawing tools to create custom characters and backgrounds. Explain what events are and apply multiple event blocks in programs. Build sequences to model digital responsibilities. Apply debugging skills to identify and correct errors. Explore how computer science can solve community problems.
Lessons	Drawing Tools: Nature Walk ■ Use ScratchJr's drawing tools to create a nature-inspired scene. Events ■ Explain events and use multiple event types in a single program. Sequences: Digital Responsibilities ■ Create a program demonstrating responsible online behavior. Hide and Seek Game ■ Use the hide block to make an interactive game.

Introduction to the Wait Block ■ Add pauses to character actions using the wait block. Introduction to Debugging

Identify and correct mistakes in sequences.

Careers in CS: Litter Free Communities

• Explore real-world applications of computer science in solving community problems.

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	 Practice creating and following sequences of instructions without a screen. Use loops to simplify repeated actions in a physical coding activity. Build teamwork and communication skills while applying coding logic. 	
Lessons	 Coding Card Game: Loops Work together to guide Scout through a maze using cards that represent movement and repeat loops. Coding Card Game: Loops 2 Tackle a new maze challenge by building efficient sequences with loops to help Scout reach the goal. 	

Unit 3: Loops (3 lessons)

Students implement loops in ScratchJr to simplify code, add rhythm, and create animations that repeat.

Objectives / Topics Covered	 Use different types of loops to repeat code. Combine loops with motion and event blocks. Develop games and animations with dynamic elements. 	
Lessons	Introduction to Repeat Loops	

Unit 4: Message Events (5 lessons)

Students learn how to make characters interact and respond to each other using message events. They explore how messages can trigger sequences, change behavior, and allow for more complex and interactive programs..

Objectives / Topics Covered	 Program characters to communicate using message events. Create interactive animations and stories with multiple characters. Use events and variables to extend program functionality.
Lessons	Introduction to Message Events

• Students will be able to use speed blocks and messages to program a character to move at different speeds.

Variables: Storing Data with Symbols

 Students will be able to model the way programs store and manipulate data using variables.

Unit 5: Pages & Grid (5 lessons)

In this unit, students build on their ScratchJr skills by learning how to use pages, grids, and special blocks to create interactive projects and games. They explore how programs can span multiple screens, move characters across grids, and include structured endings. These skills prepare students to design larger, more complex animations and activities that combine creativity with problem-solving.

Objectives / Topics Covered	 Using multiple pages in a program. Applying the "go to page" block to connect activities. Designing interactive games and greeting cards. Navigating with the grid to solve mazes. Programming endings with the "end" block.
Lessons	Pages: Create a Tapping Game ■ Design an interactive game that switches between pages when a character is tapped. Create a Mini Golf Game ■ Students will be able to use messages and loops to create a mini golf game. Digital Greeting Card ■ Students will be able to create a digital greeting card with events and loops. Grid: Solving Mazes ■ Design a maze and use grid-based movement to guide a character from start to finish. End Block: Program a Race ■ Program a race between characters and use the "end" block to stop the animation at the finish line.

Unit 6: Culmination Projects (5 lessons)

In this unit, students apply the design process to create original projects demonstrating problem-solving, personal expression, and technical skill.

Objectives / Topics Covered	 Plan, create, and revise ScratchJr projects. Use sequences, events, loops, and variables. Model real-world or imaginative scenarios. 	
Lessons	Practicing the Design Process Use the design process to solve a simple real-world problem. About Me Project (2 part lesson) Design a program that shares personal characteristics, interests, and favorite things. River Crossing Game (2 part lesson) Create an obstacle game with difficulty controlled by speed blocks.	

Unit 7: Digital Literacy (9 lessons)

In this unit, students explore essential concepts of digital literacy, including research skills, safe technology use, data storage, and networks. They also gain an introduction to artificial intelligence (AI) and machine learning by examining how computers use data to learn and make decisions. The lessons help students connect computing skills to everyday life, supporting responsible and informed use of technology.

Objectives / Topics Covered	 Conducting research and presenting findings with code. Collecting, organizing, and storing data in programs. Understanding personal and private information. Recognizing how networks and the Internet connect people. Identifying how AI tools work and what they can and cannot do. Exploring how machine learning uses data and attributes to recognize patterns.
Lessons	Guided Research (2 part lesson)

Virginia Computer Science 1st Grade Course Supplemental Materials

Resources	Description
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to computer science.
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 <u>CodeHop Resources Page</u> .	