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 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.

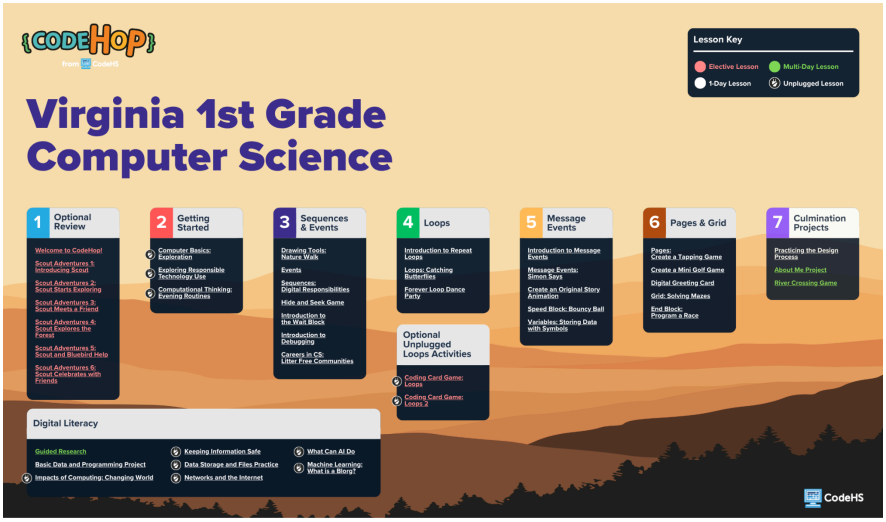
Learning Environment: This course is teacher-led and includes ready-to-use lessons following a consistent structure: Introduction, Guided Practice, Independent Practice, Extension, and Reflection. Instruction follows an "I do, we do, you do" model and incorporates spiral review to reinforce concepts 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 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 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. ● 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 programming 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 (1.CSY.1.a, 1.CSY.1.b, 1.CSY.1.c, 1.CSY.2.a, 1.CSY.2.b, 1.IC.1.a, 1.IC.1.b, 1.IC.2.a, 1.IC.3.a, 1.IC.3.d)** ● Learn how computers work and understand hardware, software, input, and output. **Exploring Responsible Technology Use (1.CYB.1.a, 1.CYB.1.b, 1.CYB.1.c, 1.CYB.1.d, 1.CYB.1.e, 1.IC.1.a, 1.IC.2.a, 1.IC.2.b)** ● Understand how to use technology safely and responsibly. **Computational Thinking: Evening Routines (1.AP.1.d)** ● 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 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 drawing tools to create a nature-inspired scene. **Events (1.AP.2.a, 1.AP.2.c)** ● Explain events and use multiple event types in a single program. **Sequences: Digital Responsibilities (1.AP.2.d)** ● 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 (1.AP.2.a, 1.AP.3.a)** ● Identify and correct mistakes in sequences. **Careers in CS: Litter Free Communities (1.IC.3.a, 1.IC.3.b, 1.IC.3.c, 1.IC.3.d)** ● 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 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 (1.AP.2.a, 1.AP.2.b)** ● Use repeat loops to make a character perform an action multiple times in a row. **Loops: Catching Butterflies (1.AP.2.b)** ● Animate butterflies using show/hide and repeat loops. **Forever Loop Dance Party (1.AP.2.a, 1.AP.2.b)** ● Use the "repeat forever" loop to create a continuous dance party animation.

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 (1.AP.2.c)** ● Program a relay race using message events. **Message Events: Simon Says (1.AP.2.c)** ● Create a “Simon Says” game where one character triggers multiple others. **Create an Original Story Animation (1.AP.2.a)**

	● Students will be able to create a program to animate an original story. **Speed Block: Bouncy Ball (1.AP.2.a)** ● Students will be able to use speed blocks and messages to program a character to move at different speeds. **Variables: Storing Data with Symbols (1.DA.1.a, 1.DA.1.b)** ● 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 programming 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 (1.AP.2.a, 1.AP.2.d)** ● Design a maze and use grid-based movement to guide a character from start to finish. **End Block: Program a Race (1.AP.2.a)** ● 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 projects. ● Use sequences, events, loops, and variables. ● Model real-world or imaginative scenarios.
Lessons	**Practicing the Design Process (1.AP.2.a, 1.AP.3.b, 1.AP.3.c)** ● Use the design process to solve a simple real-world problem. **About Me Project (2 classes)** ● Design a program that shares personal characteristics, interests, and favorite things. **River Crossing Game (2 classes)** ● 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 classes | 1.NI.1.a)** • Find information on a topic and create a visual program to present research. **Basic Data and Programming (1.DA.2.a, 1.DA.2.b, 1.DA.2.c, 1.DA.2.d)** • Collect and present data through a visual program. **Impacts of Computing: Our Community (1.IC.3.a, 1.IC.3.d)** • Identify examples of computing devices in daily life and explain how they've changed how people live, work, and play. **Keeping Information Safe (1.CYB.1.b, 1.CYB.1.c, 1.CYB.2.a, 1.CYB.2.b)** • Students will be able to identify private and personal information. **Data Storage and Files Practice (1.DA.1.a, 1.DA.1.b)** • Learn how computers store information and explore how files hold data. **Networks and the Internet (1.NI.1.b)** • Explain how networks work and how messages are shared online. **What Can AI Do? (1.IC.3.a, 1.IC.3.b, 1.IC.3.c, 1.IC.3.d)** • Identify AI tools, explain how they learn, and compare tasks better for humans vs. AI. **Machine Learning: What is a Blorg? (1.AP.1.a, 1.AP.1.b, 1.AP.1.c)** • Discover how AI recognizes patterns in data and uses attributes to make decisions.

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 **CodeHop Resources Page**.	