Virginia Computer Science Kindergarten Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Virginia Computer Science Kindergarten 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 30 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/27179/overview



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

Course Breakdown

Optional Unplugged Exploration

In this unplugged unit, students explore foundational coding concepts without using a device. Through physical activities, games, and role-play, students learn how to sequence actions and understand how events trigger responses.

Objectives / Topics Covered	● Create step-by-step sequences. ● Understand how events cause actions. ● Practice direction-following using physical movement and games. ● Develop collaboration and logical thinking skills. ● Practice mouse and keyboard usage.
Lessons	**Sequences (K.AP.2.c, K.AP.2.d, K.AP.2.e)** ● Create a sequence of step-by-step instructions for a dance to understand the importance of order. **Coding Card Game: Sequences (K.AP.2.d, K.AP.2.e)** ● Collaboratively guide Scout through a maze using sequence cards. **Coding Card Game: Sequences 2 (K.AP.2.d, K.AP.2.e)** ● Tackle a new maze together using sequence coding strategies. **Acting with Events (K.AP.2.d, K.AP.2.e)** ● Act out cause-and-effect scenarios where events trigger specific actions. **Mouse Practice (K.CSY.1.a, K.CSY.1.b)** ● Demonstrate mouse skills through dragging and clicking in interactive games. **Keyboard Introduction (K.CSY.1.a, K.CSY.1.b)** ● Explore the keyboard by identifying letters, numbers, and basic functions.

Unit 1: Getting Started (3 lessons)

Students are introduced to the CodeHop Playground, basic computer concepts, responsible technology use, and computational thinking practices. This unit provides the foundation for later lessons by establishing key routines, safety expectations, and introductory programming skills.

Objectives / Topics Covered	● Log in and navigate the CodeHop Playground. ● Identify and describe the parts of a computer. ● Explain how to use technology safely and responsibly. ● Apply computational thinking skills to everyday routines.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn to log in and explore the Playground; designed as a standalone or warm-up activity. **Computer Basics: Introduction (K.CSY.1.a, K.CSY.1.b, K.CSY.1.c, K.CSY.1.d, K.CSY.1.e, K.CSY.2.a, K.CSY.2.b, K.IC.1.a, K.IC.1.b, K.IC.2.a, K.IC.2.b, K.IC.2.c, K.NI.1.a, K.NI.1.b)** ● Discover what computers are, how they function, and how input/output devices interact. **Introduction to Responsible Technology Use (K.CSY.1.e, K.CYB.1.a, K.CYB.1.b, K.CYB.1.c, K.CYB.2.a, K.CYB.2.b, K.CYB.2.c, K.IC.1.c)** ● Learn safe and responsible behaviors with classroom technology. **Computational Thinking: Morning Routines (K.AP.2.a, K.AP.2.c, K.AP.2.d, K.AP.2.e)** ● Analyze daily routines using patterns and sequencing.

Unit 2: Programming Introduction (7 lessons)

This story-based unit introduces Scout the Squirrel and guides students through programming basics while building coding confidence through storytelling.

Objectives / Topics Covered	● Navigate the programming interface. ● Add, delete, and modify elements. ● Use motion and event blocks to animate characters. ● Explore storytelling with visual programming.
Lessons	**Scout Adventures 1: Introducing Scout** ● Explore the programming interface and add characters to begin Scout's journey. **Scout Adventures 2: Scout Starts Exploring** ● Add backgrounds and a new page as Scout begins to explore the world around them. **Scout Adventures 3: Scout Meets a Friend** ● Delete and modify characters as Scout meets a new friend during the adventure. **Scout Adventures 4: Scout Explores the Forest** ● Use motion blocks to move characters around the stage as Scout explores the forest. **Scout Adventures 5: Scout and Bluebird Help** ● Build a sequence to collect items using motion blocks. **Scout Adventures 6: Scout Celebrates with Friends** ● Create a celebratory multi-scene animation using sequences and events. **Drawing Tools: Fairy Tale Painting** ● Use painting tools to design a custom character and background.

Unit 3: Sequences & Events (5 lessons)

Students explore how sequences and events create interactivity and animation in programs, and begin using visual effects like appearing, disappearing, and size changes.

Objectives / Topics Covered	● Build sequences and use event triggers. ● Make characters appear/disappear and change size. ● Animate with motion and events.
Lessons	**Introduction to Events** ● Create a simple program that uses different types of events to control how and when characters move. **Events: Submarine Sequences (K.AP.2.c, K.AP.2.d, K.AP.2.e)** ● Use event and motion blocks to program a character's movement in an animated underwater scene. **Introduction to Show and Hide Blocks (2 classes)** ● Add show and hide blocks to make characters appear and disappear within a program. **Introduction to Grow and Shrink Blocks** ● Create an animated scene where characters change size using grow and shrink blocks.

Unit 4: Pages (3 lessons)

Students expand their storytelling by using multi-page projects, learning how to control scene transitions with navigation blocks.

Objectives / Topics Covered	● Create multi-page programs with different scenes. ● Use the "go to page" block to switch between pages within a project. ● Develop storytelling skills through sequencing and scene transitions. ● Explain how technology supports careers such as spacecraft design.
Lessons	**Pages: Dragon Story (K.AP.2.b)** ● Create a multi-page story featuring a dragon, with different actions and scenes on each page. **Using the Go To Page Block** ● Build a program that switches between pages using the "go to page" block for interactive navigation.

	Careers in CS: Spacecraft Design ● Explain how technology benefits careers and animate an astronaut and spacecraft in a space program.

Unit 5: Block Exploration (6 lessons)

Students build confidence with foundational programming blocks through hands-on exploration. They practice using motion, sound, and loop blocks to create interactive projects. Debugging is introduced to help students identify and fix errors in their code. By the end of the unit, students will understand how different types of blocks can be combined to create simple programs and animations.

Objectives / Topics Covered	● Explore speed, sound, and say blocks in interactive projects. ● Share personal information through programming. ● Debug motion and event-based sequences. ● Use loops to repeat actions and simplify programs.
Lessons	**Transportation Speeds and Sounds** ● Combine sound and speed blocks to animate different types of transportation. **All About Me! (2 classes)** ● Design a program that shares favorite things and personal interests. **Debugging: Events and Motion** ● Find and fix bugs in a program that uses event and motion blocks. **Loops** ● Use loops and explain how loops are used to repeat code. **Loops: Frog and Rabbit** ● Practice using loops to repeat movements in an interactive animation with animals.

Unit 6: Culmination Projects (5 lessons)

Students apply all coding concepts in open-ended projects, showcasing creativity, design thinking, and problem-solving.

Objectives / Topics Covered	● Use sequences, events, and loops in original projects. ● Apply the design process to create interactive experiences. ● Express creativity through animation and storytelling.
Lessons	**Introduction to the Design Process (K.AP.2.c, K.AP.2.e)** ● Use structured steps to solve a simple problem through coding. **Chicken Crossing Game (2 classes)** ● Create an interactive game with adjustable speed and timing. **Wildlife Scene Project (2 classes)** ● Animate a nature scene using all core programming concepts.

Unit 7: Digital Literacy (8 lessons)

Students are introduced to foundational digital literacy skills that extend beyond coding. They will learn how information is stored, shared, and analyzed, as well as how artificial intelligence systems make decisions and support human activities. Lessons emphasize research, networks, data, and the responsible use of digital tools.

Objectives / Topics Covered	● Conduct basic online research and communicate findings. ● Collect, organize, and analyze real-world data. ● Understand how computers store data in files. ● Explore how networks connect people and devices. ● Model how AI uses decision trees to make choices.

	● Identify practical ways AI helps solve problems and impacts jobs.
Lesson	**Introduction to Research (2 classes | K.CSY.1.d, K.DA.1.a, K.DA.2.c)** ● Use safe research tools to find information on a topic and create a simple visual program to present what was learned. **What Can Data Tell Us? (K.DA.1.a, K.DA.1.b, K.DA.1.c, K.DA.2.a, K.DA.2.b, K.DA.2.c)** ● Collect and analyze transportation data using charts. **Impacts of Computing: Daily Life (K.IC.1.a, K.IC.1.b, K.IC.2.a, K.IC.2.b, K.IC.2.c)** ● Identify examples of computing devices in daily life and explain how they've changed how people live, work, and play. **Introduction to Data Storage and Files (K.CSY.1.b)** ● Understand how data is stored as files on digital devices. **Using Networks to Connect (K.NI.1.a, K.NI.1.b)** ● Understand how people and devices connect with and without wires, and how networks enable communication. **Sorting with Decision Trees (K.AP.1.a, K.AP.1.b, K.AP.1.c)** ● Understand how AI uses data to sort and learn. **How AI Helps Us (K.IC.1.a, K.IC.1.b, K.IC.2.a, K.IC.2.b, K.IC.2.c)** ● Learn how AI assistants work and support real-life tasks.

Virginia Computer Science Kindergarten 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**.	