Virginia Computer Science
2nd Grade Course Syllabus
One Year for Elementary School, 36 Hours

Course Overview and Goals

The Virginia Computer Science 2nd 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.

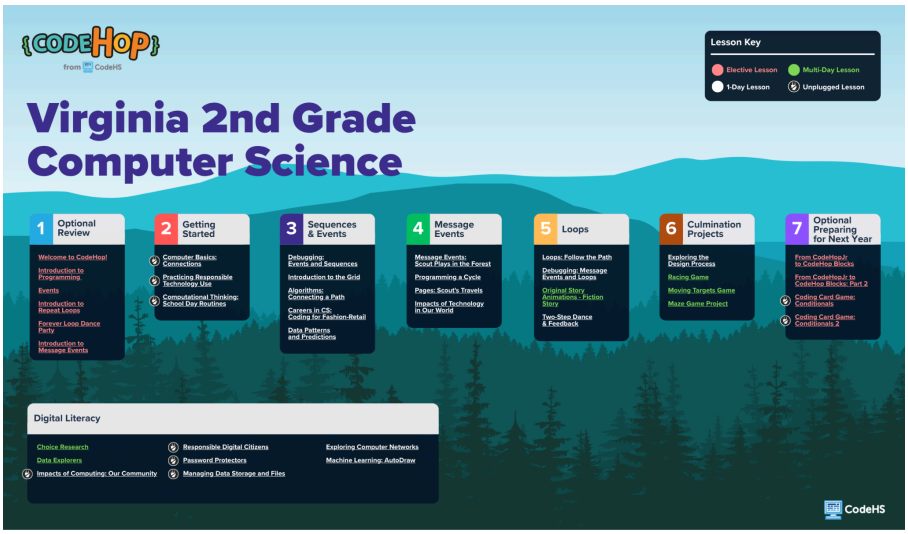
Learning Environment: This course is teacher-led and includes ready-to-use lessons following a consistent structure: Introduction, Guided Practice, Independent Practice, Extension, and Reflection. Instruction follows an "I do, we do, you do" model and incorporates spiral review to reinforce concepts 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/27181/overview



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

Course Breakdown

Optional Review

This optional review unit helps students refresh foundational skills such as navigating the Playground and understanding events and loops.

Objectives / Topics Covered	● Review basic programming interface and functionality. ● Practice using events, loops, and messages. ● Reinforce foundational computer science concepts before progressing.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Introductory lesson to help students log in and explore the CodeHS Playground; ideal as a warm-up or standalone activity. **Introduction to Programming** ● Navigate the programming interface and create a scene with characters and background. **Events** ● Use multiple event blocks and explain how events work in programming. **Introduction to Repeat Loops** ● Learn how to use repeat loops to make actions happen more than once in a row. (2.AP.2.b, 2.AP.2.c) **Forever Loop Dance Party** ● Use “repeat forever” loops to build a looping dance animation. **Introduction to Message Events** ● Program a relay race that uses messages to cause characters to interact.

Unit 1: Getting Started (3 lessons)

Students are introduced to computers, responsible technology use, and computational thinking. They learn what computers are, how to use them safely, and how to break down everyday routines into steps, patterns, and sequences.

Objectives / Topics Covered	● Learn basic computer components and how they work together. ● Demonstrate safe and responsible technology use. ● Apply computational thinking to familiar routines.
Lessons	**Computer Basics: Connections (2.CSY.1.a, 2.CSY.1.b, 2.CSY.1.c, 2.CSY.2.a, 2.CSY.2.b, 2.IC.1.a, 2.IC.3.a, 2.IC.3.b, 2.IC.3.c)** ● Learn what a computer is and how hardware, software, input, and output work together. **Practicing Responsible Technology Use (2.CYB.1.a, 2.IC.1.b, 2.IC.1.c, 2.IC.2.a, 2.IC.2.b)** ● Demonstrate how to safely and respectfully use digital tools. **Computational Thinking: School Day Routines (2.AP.2.a)** ● Break down school day routines using patterns, sequencing, and problem-solving skills.

Unit 2: Sequences & Events (5 lessons)

In this unit, students dive deeper into programming logic with sequences, events, and debugging while beginning to use grid-based movement and algorithm design.

Objectives / Topics Covered	● Program sequences and trigger actions with events. ● Debug code and refine algorithms. ● Use the grid and character size to plan movement.

Lessons	**Introduction to the Grid** ● Use the grid to move characters precisely on the stage. **Debugging: Events and Sequences** ● Find and fix errors in sample code to improve program outcomes. **Algorithms: Connecting a Path (2.AP.3.c)** ● Write and adjust step-by-step instructions to connect characters on the screen. **Careers in CS: Coding for Fashion-Retail (2.IC.3.a, 2.IC.3.b, 2.IC.3.c)** ● Learn how coding is used in fashion, then animate a custom fashion character. **Data Patterns and Predictions (2.DA.1.c, 2.DA.2.b, 2.DA.2.c)** ● Analyze patterns in data and program an animation to communicate insights.

Unit 3: Message Events (3 lessons)

Students explore how messages can control program flow and build animations that model real-world systems and storytelling.

Objectives / Topics Covered	● Program interactions between characters using message events. ● Model cycles and communication flows. ● Coordinate scenes with messaging across pages.
Lessons	**Message Events: Scout Plays in the Forest** ● Control a program's flow using messages to trigger actions. **Programming a Cycle** ● Use messages to represent and loop through a natural or repeating cycle. **Pages: Scout's Travels** ● Navigate between pages using messages to help Scout move through a story.

Unit 4: Loops (5 lessons)

Students will explore how loops simplify patterns in code, practice debugging, build timers, and create animated stories. They'll also learn to revise programs using peer feedback and give proper credit.

Objectives / Topics Covered	● Identify and use loops to repeat actions and patterns. ● Debug code involving message events and loops. ● Use loops with timing and motion blocks. ● Create and revise animations with feedback and attribution.
Lessons	**Loops: Follow the Path (2.AP.2.b, 2.AP.2.c)** ● Identify repeated patterns and use loops to move characters efficiently. **Debugging: Message Events and Loops (2.AP.3.b)** ● Find and fix coding errors involving loops and message events. **Create an Original Story Animation (2 classes)** ● Design and animate a unique story using characters, loops, and events. **Two-Step Dance & Feedback** ● Code a dance, revise it with peer feedback, and give credit to collaborators.

Unit 5: Culmination Projects (9 lessons)

Students apply everything they've learned to build creative, interactive programs. These open-ended projects reinforce the design process, interactivity, and peer review.

Objectives / Topics Covered	● Plan and revise programs using the design process. ● Build original games with events, messages, loops, and pages. ● Collaborate and revise work based on feedback.

Lessons	**Exploring the Design Process (2.AP.2.a, 2.AP.3.a, 2.AP.3.d, 2.AP.3.e)** ● Apply all steps of the design process independently to create and improve a program that includes loops to solve a real-world problem from a user's perspective. **Racing Game (2 classes | 2.AP.2.c)** ● Create a racing game using events, loops, and messages to control character actions. **Moving Targets Game (3 classes)** ● Program a game with moving targets that uses sequences, events, and pages. **Maze Game Project (3 classes)** ● Design a maze game and revise it using feedback while applying key programming concepts.

Unit 6: Digital Literacy (11 lessons)

Students explore key aspects of digital citizenship, online safety, data handling, and the role of artificial intelligence. They learn how to protect personal information, manage files, and critically evaluate how technology and AI are used in everyday life.

Objectives / Topics Covered	● Collect, assess, and present research using digital tools. ● Develop and visualize data through surveys and programs. ● Practice safe, responsible digital citizenship. ● Use strong usernames and passwords to protect information. ● Understand file storage and computer organization. ● Explain how networks connect devices and share data. ● Identify how AI recognizes patterns and supports human tasks.
Lessons	**Choice Research (2 classes | 2.NI.1.b, 2.NI.1.c)** ● Research a self-selected topic using reliable sources and create a visual program to share findings. **Data Explorers (2 classes)** ● Develop a survey, collect data, and present results through an interactive program. **Impacts of Computing: Changing World (2.IC.3.a, 2.IC.3.b, 2.IC.3.c)** ● Identify examples of computing devices in daily life and explain how they've changed how people live, work, and play. **Responsible Digital Citizens (2.CYB.1.a, 2.CYB.1.b, 2.CYB.1.c, 2.CYB.2.a)** ● Explore digital footprints, cyberbullying, and how to report unsafe online behavior. **Password Protectors (2.CYB.2.a, 2.CYB.2.b, 2.CYB.2.c)** ● Explore the importance of usernames and passwords and practice strategies to keep them secure. **Managing Data Storage and Files (2.CSY.1.d)** ● Recognize how computers store data and how it is organized. **Exploring Computer Networks (2.NI.1.a)** ● Model how devices send and receive information over a network. **Machine Learning: AutoDraw (2.AP.1.a, 2.AP.1.b, 2.AP.1.c)** ● Learn how AI recognizes patterns in drawings and practice sorting, classifying, and organizing data.

Optional Preparing for Next Year

These optional lessons help students build on what they've learned by applying conditionals in unplugged activities and transitioning from CodeHopJr to CodeHop blocks.

Objectives / Topics Covered	● Use conditionals to create step-by-step instructions. ● Collaborate to solve maze challenges with logic. ● Explore the CodeHop interface and basic programming tools. ● Create simple programs using events and loops.

Lessons	**From CodeHopJr to CodeHop Blocks** ● Learn the basics of CodeHop by building a simple program. **From CodeHopJr to CodeHop Blocks: Part 2** ● Create a program that includes an event and a repeating action using a loop. **Coding Card Game: Conditionals** ● Use conditionals to guide Scout through a maze using a card-based activity. **Coding Card Game: Conditionals 2** ● Continue practicing conditional logic with new maze challenges.

Virginia Computer Science 2nd 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**.	