

Virginia Computer Science 3rd Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Virginia Computer Science 3rd Grade Course** introduces students to foundational programming concepts through **Scratch**, 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 **Scratch** 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/27182/overview



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

Course Breakdown

Unit 1: Getting Started (3 lessons)

Students build foundational computer science and technology skills by learning about computing systems, responsible technology use, and computational thinking. These lessons prepare students to log in, navigate, and approach problem-solving with structured processes.

Objectives / Topics Covered	 Identify components of a computing system and basic troubleshooting. Demonstrate safe and responsible technology use. Apply computational thinking to break down and sequence real-world routines.
Lessons	Welcome to CodeHop! (15 minute lesson) ■ Practice logging in and exploring the Playground before starting a full lesson. Introduction to Computing Systems ■ Identify parts of the computing system and solve basic hardware/software problems. Safe and Responsible Use of Technology ■ Create a personal plan for balanced and safe technology use. Computational Thinking: Design an Obstacle Course ■ Create a program where multiple sequences run at the same time.

Unit 2: Scratch Exploration (5 lessons)

In this story-driven unit, students follow Scout the Squirrel on an adventure through Scratch while learning to animate characters, create stories, and use basic coding concepts.

Objectives / Topics Covered	 Explore the Scratch interface and basic block types. Animate characters with motion, looks, and speech. Use sequences, events, and loops to build short stories. Customize characters and scenes with drawing tools.
Lessons	Scout's Scratch Expedition Part 1

Unit 3: Sequences & Events (8 lessons)

Students expand their programming skills by combining sequences, events, and broadcast messages to create interactive stories, models, and animations. They will also practice remixing existing projects and connect coding to real-world applications.

Objectives / Topics	 Build and run parallel sequences. Use events and broadcast messages to control sprite actions.
Covered	Create models of real-world processes with coding.
	 Plan, animate, and remix interactive stories. Connect coding skills to real-world applications and careers.

Lessons	Sequences: Parallel Programming • Create a program where multiple sequences run at the same time.
	Events
	 Use event blocks to start actions when keys are pressed or sprites are clicked.
	Broadcast Messages: Marco Polo
	 Send and receive messages between sprites to coordinate actions.
	Modeling Life Cycles
	 Animate a frog's life cycle to model real-world processes.
	Creative Storytelling (2 day lesson)
	 Plan and animate a story using sequences and event-based interactions.
	Remixing Programs
	 Remix digital programs and give appropriate credit to original creators.
	Careers in CS: Health and Fitness
	 Explore how coding supports fitness and create an animation showing technology in action.

Unit 4: Loops & Conditionals (5 lessons)

Students build on their foundational coding skills by learning how to use loops to repeat instructions and conditionals to make decisions in programs. They practice debugging, explore increasingly complex conditional logic, and create interactive projects that respond to different inputs.

Objectives / Topics Covered	 Explain and apply loops to repeat instructions. Debug programs with loops and events. Use simple if/then conditionals in Scratch. Create programs that apply conditional logic for decision-making. Explore if/then/else conditionals to build complex outcomes.
Lessons	Loops

Unit 5: Variables & Functions (5 lessons)

In this unit, students use variables, lists, comparison operators, and custom functions to manage and organize data in more complex programs.

Objectives / Topics Covered	 Create and update variables in Scratch. Use comparison operators in decision-making. Build lists to organize data for games and activities. Define and reuse functions to simplify large programs.
Lessons	Variables ■ Learn what a variable is and how to create and change its value in Scratch. Introduction to Comparison Operators ■ Use comparison operators with numbers and variables in if/else logic. I-Spy with Lists (2 day lesson) ■ Build an interactive I-Spy game using lists and variables to track items and responses. Introduction to Functions

• Learn how to define and use functions to simplify and organize code.

Unit 6: Culmination Project (4 lessons)

Students apply their programming knowledge and creativity to larger projects that integrate design thinking, data, and storytelling. These culminating lessons encourage students to combine skills from across the course to demonstrate mastery and problem-solving.

Objectives / Topics Covered	 Apply the design thinking process to improve accessibility. Use the inquiry process to collect, analyze, and present data. Combine multiple coding skills into capstone-style projects.
Lessons	 Introduction to Design Thinking (2 day lesson) ◆ Use the design thinking process to brainstorm, build, and improve a program. Inquiry Project: Survey Bar Graph (2 day lesson) ◆ Follow the inquiry process to investigate a topic and modify a Scratch program to display survey results using a bar graph.

Unit 7: Digital Literacy (6 lessons)

Students build foundational digital literacy skills by exploring safe and responsible technology use, data evaluation, cybersecurity, and networks. They also consider how technology impacts culture and practice attribution when using online information.

Objectives / Topics Covered	 Search for information online and give proper credit. Evaluate data for reliability, analyze results, and make predictions. Create and protect strong usernames and passwords. Identify basic cybersecurity threats and safety practices. Model how data travels across networks. Reflect on the cultural impacts of technology over time.
Lessons	What Can I Use Online? • Practice safe online research, use keywords effectively, and cite sources properly. Data Detectives • Evaluate and analyze data for reliability to draw conclusions and make predictions. Strong Usernames and Passwords • Create strong credentials and explain how they keep information safe. Scout's Cybersecurity Adventure: Part 1 • Identify common cyber threats and strategies for staying safe online. Modeling Network Connections • Show how data travels between devices using a network. Impacts of Computing: Introduction • Explore how computing has changed communities and evolved over time.

Virginia Computer Science 3rd 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.
Design-Your-Own-Lesson Scratch Templates	Empower your students to explore and express their knowledge creatively with our versatile Scratch graphic organizer templates. Designed with adaptability and ease of use in mind, these interactive tools transform any subject into an engaging, hands-on learning experience.
All of these resources and more are found on the <u>CodeHop Resources Page</u> .	