CodeHS

Georgia Computer Science Standards of Excellence: 5th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Georgia Computer Science Standards of Excellence: 5th Grade** 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**, each approximately 45 minutes long. This provides a full school year of material if teaching one lesson per week. Digital literacy lessons are also available to complement the programming curriculum with non-programming computer and technology skills. Additionally, this course includes optional interdisciplinary lessons in math, science, ELA, and social studies to support cross-curricular integration.

Programming Environment: Students will write and run programs in **Scratch** embedded and saved in the CodeHS 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/26343/overview?lang=en.

Course Breakdown

Unit 1: Optional Review (4 weeks)

In this optional unit, students revisit foundational skills in Scratch and computer science. They practice navigating the CodeHS platform, define key vocabulary, and create animations using coordinates and custom artwork.

Objectives / Topics Covered	● Log in and navigate the CodeHS Playground. ● Define computer science vocabulary and apply it in Scratch. ● Use the coordinate plane for sprite movement and positioning. ● Customize sprites and backdrops using Scratch drawing tools.
Lessons	**Welcome to CodeHS!** ● Learn how to log in and use the CodeHS Playground. This short introductory lesson can be used on its own, or right before a full lesson. **Introduction to Computer Science and Scratch** ● Define important computer science vocabulary and create a simple program in Scratch. **The Coordinate Plane** ● Create an open-ended animation using the coordinate plane in Scratch. **Scratch Drawing Tools** ● Create customized sprites and backdrops using the drawing tools.

Unit 2: Getting Started (2 weeks)

In this unit, students strengthen their understanding of computing systems and online behavior. They identify tech components and explore the impact of their words and actions in digital spaces.

Objectives / Topics Covered	● Identify hardware and software in computing systems. ● Troubleshoot simple computer issues. ● Understand cyberbullying, digital property, and being an upstander online.
Lessons	**Practicing with Computing Systems** ● Identify parts of the computing system and identify simple hardware and software problems. **Our Words Have Power (Cyberbullying)** ● Explain what cyberbullying is, how it affects others, how to be an upstander, and that work developed online is the property of the creator.

Unit 3: Sequences & Events (5 weeks)

Students build programs using events and sequences while strengthening debugging skills and connecting coding to real-world applications.

Objectives / Topics Covered	● Apply computational thinking to structure multi-step programs. ● Debug programs by decomposing and correcting code. ● Compare and refine algorithms. ● Connect coding concepts to real-world careers through programming.
Lessons	**Computational Thinking: Design a Neighborhood** ● Use computational thinking to design a neighborhood. **Events: Traveling with Scout** ● Use events in a program. **Debugging: Make a Pizza** ● Decompose a program to debug and make the program run as intended. **Careers in CS: Major League Baseball**

	• Explain how coding can be used in sports and retell events from an article. **Compare and Refine Algorithms** • Analyze multiple algorithms for a task to determine which is the most appropriate.

Unit 4: Loops (3 weeks)
In this unit, students use loops to build efficient, creative programs in Scratch. They explore visual effects, multi-scene animations, and pattern-based designs using repeat structures.

Objectives / Topics Covered	• Use repeat loops to simplify repetitive actions in a program. • Create visual animations and scene transitions using loops. • Combine drawing tools and loops to generate geometric designs.
Lessons	**Creating Turtle Graphics** • Use the pen tool in Scratch to create looping turtle graphics. **Animation Loops Project (2-part lesson)** • Use repeat loop blocks to program an animation with multiple scenes.

Unit 5: Conditionals & Operators (4 weeks)
In this unit, students deepen their understanding of conditional logic and operators to build interactive, decision-driven programs. They apply planning and decomposition to break complex projects into manageable parts.

Objectives / Topics Covered	• Use if/then statements and operators to control program logic. • Apply planning and decomposition to design interactive programs. • Combine conditionals, variables, and events to simulate real-world behaviors.
Lessons	**Conditionals: Mazes** • Create a program that uses conditionals. **Plan a Quest (2-part lesson)** • Plan and decompose the steps needed to create a quest program. **Operators: Coin Flip** • Create a coin flipping program using variables and operators.

Unit 6: Variables & Lists (4 weeks)
In this unit, students manage and track information using variables and lists. They build interactive simulations and games that demonstrate how data affects program behavior.

Objectives / Topics Covered	• Use variables to control elements such as speed, pitch, and score. • Create lists to store multiple values in interactive programs. • Combine variables, lists, and operators to simulate real-world choices.
Lessons	**Variables in Dance** • Use variables to control pitch and dance speeds in a program. **Game Mechanics with Comparison Operators** • Use comparison operators and variables to create ending game mechanics. **Lists: Shopping with Scout (2-part lesson)** • Create a shopping simulator using variables, lists, and operators.

Unit 7: Clones & Functions (6 weeks)
In this unit, students explore advanced programming patterns using clones, functions, and object-oriented logic. They build interactive games with repeated elements and customizable behavior, while also learning to reuse code efficiently.

Objectives / Topics Covered	• Use clones to create repeated actions and sprites. • Apply functions with input to organize and reuse code. • Simulate objects with changing characteristics using variables and randomness. • Build games that use functions, inputs, and clones to support gameplay.
Lessons	**Clones: Throwing Acorns Game** • Create a throwing acorns game using clones. **Clones in Games (2-part lesson)** • Use clones to program an endless runner game and explain why clones are useful in game programs. **Classes and Objects in Games (2-part lesson)** • Learn about classes and objects in programming while creating an interactive game and using randomizers to change the characteristics of objects. **Functions: About Me** • Create and use a function with input in a program.

Unit 8: Culmination Projects (8 weeks)

Students apply a wide range of programming skills in open-ended projects that highlight creativity, problem-solving, and real-world applications. They plan, design, and build multi-concept programs aligned with math and app design.

Objectives / Topics Covered	• Combine loops, conditionals, variables, and events in original projects. • Use functions to structure interactive programs with real-world applications. • Design user-centered apps and games using the design thinking process. • Apply math concepts like area and perimeter through code.
Lessons	**Game Design Project (3-part lesson)** • Design and create a game using multiple programming skills such as loops, conditionals, and variables. **House Design with Area and Perimeter (2-part lesson)** • Calculate and use the area and perimeter of a room to create a house design using functions. This version of the lesson is focused on Computer Science concepts. **Design an App (3-part lesson)** • Use the design thinking process to design an app that helps to solve a user's need.

Unit 9: Digital Literacy (4 weeks)

In this unit, students examine how technology affects data, learning, safety, and research. They engage with data visualization,consider the impact of computing, and learn laws and policies that protect digital citizens.

Objectives / Topics Covered	• Visualize and interpret data to support a claim. • Discuss impacts of computing on society. • Explore cybersecurity laws and local policies. • Strengthen digital research and information-use habits.
Lessons	**Use and Search the Right Way** • *This lesson is coming soon!* **Using Digital Tools to Create Line Graphs** • Examine a table of information and convert the values into a data visualization (line graph) that supports a claim. **Impacts of Computing: Innovation** • *This lesson is coming soon!* **Cybersecurity Policies and Laws** • Explain policies and how they relate to their classroom or school, and research and explain a cybersecurity law specific to their state.

Unit 10: Optional Interdisciplinary (9 weeks)
In this cross-curricular unit, students apply computer science skills to explore core concepts in math, science, ELA, and social studies. They create interactive simulations, games, and animations that reinforce academic learning and demonstrate mastery through creative coding.

Objectives / Topics Covered	● Use loops, conditionals, variables, and messages to model academic content. ● Reinforce math skills like decimal operations and fractions through coding. ● Simulate scientific processes with interactive animations. ● Apply programming to demonstrate grammar, reading fluency, and historical understanding.
Lessons	**Decimal Multiplication and Conditionals** ● Use if/then conditionals to review multiplication with decimals. **Add and Subtract Fractions** ● Use broadcast messages and comparison operators to create a fractions quiz game. Recognize and use patterns in the program. **Constructive and Destructive Processes** ● Create an animation that models how volcanoes change surface features through a constructive process. **Cycle of Matter** ● Use events and messages to create an animated model of the cycle of matter. **Punctuate a Title** ● Create a game using conditionals and operators to demonstrate understanding of punctuation in titles. **Nonfiction Animated Recordings** ● Use events to create a clear, animated reading of a nonfiction text. **Interactive Map of the 13 Colonies (2-part lesson)** ● Use events, conditionals, and variables to create an interactive map of the 13 colonies. Break a large program into smaller tasks to ease development. **Revolutionary War Timeline** ● Create and control an interactive timeline using inputs, events, conditionals, and variables.

5th Grade Course Supplemental Materials

Resources	**Description**
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to computer science with CodeHS.
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 &	This guides students in engaging with computational thinking concepts,

Computational Thinking (Spanish)	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.
These resources and more are found on the **Elementary Resources Page**.	