CodeHop

Georgia Computer Science 3rd Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Georgia Computer Science 3rd 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. Digital literacy lessons are also available to complement the programming curriculum with non-programming skills. Additionally, this course includes optional interdisciplinary lessons to support cross-curricular integration.

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.

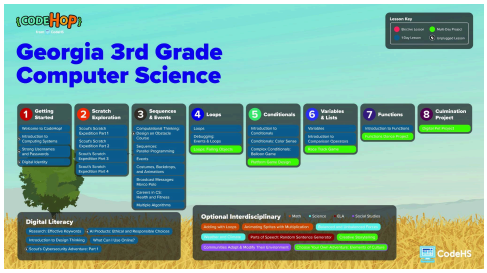
The lessons are delivered in an **"I do, we do, you do"** format, ensuring a gradual release of responsibility and fostering confidence in students as they learn. The concepts taught in this course spiral across grade levels, ensuring that students can build upon their understanding year after year. The course includes approximately **36 instructional hours**, providing a full school year of material if teaching one lesson per week. Depending on your pacing, you may find opportunities to adjust 1–2 lessons to best fit your calendar. Lesson prep notes on each lesson page provide ideas for streamlining activities while maintaining key learning goals.

Programming Environment: Students will write and run programs in the CodeHop platform. The environment supports interactive, hands-on programming, enabling students to create and debug projects in a user-friendly interface.

Assessments: Some units conclude with a brief five-question quiz designed to assess student understanding of the concepts and skills introduced in the unit. Quizzes can be assigned digitally, completed as a QuizBuzz activity, or printed for paper-based completion.

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/26335/overview?lang=en.



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

Course Breakdown

Unit 1: Getting Started (3 weeks)

In this optional unit, students will review how to log in and navigate the CodeHop Playground, identify parts of the computing system, and demonstrate safe digital practices by creating strong passwords and recognizing the connection between their real and online identities.

Objectives / Topics Covered	● Log in and navigate the CodeHop Playground. ● Identify parts of a computing system and basic troubleshooting strategies. ● Practice safe digital habits through strong passwords and positive online behavior.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and use the CodeHop Playground. This short introductory lesson can be used on its own, or right before a full lesson. **Introduction to Computing Systems (CSS.EL.3-5.1, CSS.EL.3-5.1.1, CSS.EL.3-5.1.2, CSS.EL.3-5.1.3)** ● Identify parts of the computing system and solve simple hardware and software problems. **Strong Usernames and Passwords (CSS.DC.3-5.3)** ● Develop a strong username and password and explain how a strong password keeps their information safe. **Digital Identity (CSS.DC.3-5.3.3, CSS.DA.3-5.9)** ● Connect their real world identity with their online identity and identify actions that create a positive digital footprint. **Getting Started Quiz (5 questions)**

Unit 2: Programming Exploration (4 weeks)

In this story-driven unit, students build foundational skills by creating animated scenes with Scout. They explore motion, looks, events, and loops while developing interactive stories that reflect creativity and sequential thinking.

Objectives / Topics Covered	● Use basic blocks to program movement and dialogue. ● Add and animate sprites using sequences, events, and loops. ● Build interactive stories using motion, looks, and event blocks. ● Apply storytelling skills through coding in a structured narrative.
Lessons	**Scout's Programming Expedition Part 1 (CSS.CT.3-5.5.3)** ● Use basic commands to program a sprite to move and talk. **Scout's Programming Expedition Part 2 (CSS.CT.3-5.5.3)** ● Follow a story and add sprites to create a sequence for animation. **Scout's Programming Expedition Part 3 (CSS.CT.3-5.5.3)** ● Create an animated Scout story using loops, events, looks, and motion blocks. **Scout's Programming Expedition Part 4 (CSS.CT.3-5.5.3)** ● Create an animated Scout story using events, looks, and motion blocks. **Programming Exploration Quiz (5 questions)**

Unit 3: Sequences & Events (7 weeks)

In this unit, students expand their understanding of sequences, events, and algorithms while designing interactive animations and programs. They explore parallel programming, use broadcast messages for sprite interaction, and apply computational thinking to real-world contexts.

Objectives / Topics Covered	● Use computational thinking to design and sequence multi-step tasks. ● Create programs with parallel sequences, events, and animated backdrops. ● Trigger sprite interactions using broadcast messages. ● Compare multiple algorithms to solve a problem efficiently. ● Explore how technology supports health and fitness through animation.
Lessons	**Computational Thinking: Design an Obstacle Course** ● Use Computational Thinking to design an obstacle course. **Sequences: Parallel Programming (CSS.CT.3-5.5.2, CSS.CT.3-5.5.3)** ● Create a program using parallel sequences. **Events (CSS.CT.3-5.5.4)** ● Create a program using events. **Costumes, Backdrops, and Animations (CSS.CT.3-5.5.3, CSS.CT.3-5.5.4)** ● Create a program that includes animated sprites and interactive backdrops. **Broadcast Messages: Marco Polo (CSS.CT.3-5.5.2, CSS.CT.3-5.5.3, CSS.CT.3-5.5.4)** ● Use broadcast messages to trigger action between sprites in a program. **Careers in CS: Health and Fitness (CSS.KC.3-5.2.2, CSS.KC.3-5.2.3, CSS.CT.3-5.5.3)** ● Explain how fitness and coding can amplify human strengths and create an animation to demonstrate how technology can be used in health and fitness. **Multiple Algorithms (CSS.IDC.3-5.4.3, CSS.CT.3-5.5.1, CSS.CT.3-5.5.4)** ● Program multiple algorithms to complete a task, and compare algorithms to assess which one best meets their needs. **Sequences & Events Quiz (5 questions)**

Unit 4: Loops (4 weeks)

In this unit, students deepen their understanding of loops by using them to simplify repetitive tasks. They explore different types of loops, debug programs with events and loops, and compare loop algorithms to choose the most effective solution.

Objectives / Topics Covered	● Explain and apply loops to repeat actions in a program. ● Debug programs involving events and loops by breaking them into smaller parts. ● Compare different loop types to determine the most efficient solution.
Lessons	**Loops (CSS.CT.3-5.5.3, CSS.CT.3-5.5.4)** ● Explain that a loop repeats one or more instructions, and use loops. **Debugging: Events and Loops (CSS.CT.3-5.5.3)** ● Decompose a program to debug and make the program run as intended. **Loops: Falling Objects (2 classes | CSS.IDC.3-5.4.3, CSS.CT.3-5.5.3, CSS.CC.3-5.6.1)** ● Create a program using different loops and compare the advantages and disadvantages of each loop algorithm. **Loops Quiz (5 questions)**

Unit 5: Conditionals (5 weeks)

In this unit, students explore conditional logic to make programs more interactive and responsive. They use if/then and if/then/else blocks to control behavior, apply conditionals in game design, and practice debugging to refine their projects.

Objectives / Topics Covered	● Use if/then and if/then/else blocks to control program behavior. ● Apply conditionals to create interactive animations and games. ● Combine loops, events, and conditionals to build more complex programs. ● Debug and improve programs that rely on condition-based logic.
Lessons	**Introduction to Conditionals (CSS.CT.3-5.5.4)** ● Explain what a conditional is in programming and create a program with if/then blocks.

	Conditionals: Color Sense (CSS.CT.3-5.5.4) ● Explain what a conditional is and use them in a program. **Complex Conditionals: Balloon Game (CSS.CT.3-5.5.4)** ● Explain what an "if/then/else" conditional is and use it in a program. **Platform Game Design (2 classes | CSS.KC.3-5.2, CSS.IDC.3-5.4, CSS.IDC.3-5.4.1, CSS.IDC.3-5.4.2, CSS.CT.3-5.5, CSS.CT.3-5.5.3, CSS.CT.3-5.5.4, CSS.CC.3-5.6.1, CSS.GC.3-5.7.2, CSS.GC.3-5.7.5, CSS.DA.3-5.9.2)** ● Implement platform game mechanics using keyboard events, loops, and conditionals; debug and improve the program. **Conditionals Quiz (5 questions)**

Unit 6: Variables & Lists (4 weeks)

In this unit, students learn how to store, update, and compare data using variables and lists. They use comparison operators in conditionals and apply these concepts in a game that tracks scores and player choices.

Objectives / Topics Covered	● Define and use variables to store and change data. ● Use comparison operators within conditionals to control program logic. ● Create interactive programs that track scores using variables and lists.
Lessons	**Variables (CSS.CT.3-5.5.4)** ● Explain what a variable is and create and change the value of a variable in a program. **Introduction to Comparison Operators (CSS.CT.3-5.5.4)** ● Use comparison operators as the condition in an if/else block, and use numbers and variables as values in a comparison operator. **Race Track Game (2 classes | CSS.CT.3-5.5.4)** ● Use drawing tools to design a race car and track, then program a racing game that tracks scores using conditionals, variables, and lists. **Variables & Lists Quiz (5 questions)**

Unit 7: Functions (4 weeks)

In this unit, students are introduced to functions as a way to organize and reuse code. They apply functions to simplify sequences and create structured, repeatable actions in interactive projects.

Objectives / Topics Covered	● Define and use functions to group related commands. ● Call functions to perform actions in a specific sequence. ● Apply functions creatively in music- and movement-based programs.
Lessons	**Introduction to Functions (CSS.CT.3-5.5.3, CSS.CT.3-5.5.4)** ● Create and use functions in a program. **Functions Dance Project (3 classes | CSS.CT.3-5.5.2, CSS.CT.3-5.5.4)** ● Create and use functions to call dance moves in a sequence that aligns with the music. **Functions Quiz (4 questions)**

Unit 8: Culmination Projects (3 weeks)

In this unit, students apply their full range of programming skills to design an interactive digital pet. They integrate events, conditionals, variables, and broadcasts to bring their creations to life and demonstrate mastery of core computer science concept

Objectives / Topics Covered	● Combine events, conditionals, variables, and broadcasts in a single project. ● Use comparison operators to control interactions and behavior. ● Design and program a digital pet with interactive features.

Lessons	**Digital Pet Project (3 classes | CSS.IDC.3-5.4.3, CSS.CT.3-5.5.2, CSS.CT.3-5.5.3, CSS.CT.3-5.5.4, CSS.CC.3-5.6.1, CSS.GC.3-5.7.2, CSS.GC.3-5.7.5)** ● Create a digital pet project using events, conditionals, variables, comparison operators, and broadcasts.

Unit 9: Digital Literacy (6 weeks)

In this unit, students develop responsible digital habits, explore impacts of computing, and learn how to stay safe online. They practice evaluating sources, explore ethical technology use, and are introduced to foundational concepts in cybersecurity and design thinking.

Objectives / Topics Covered	● Evaluate search keywords and assess the credibility of sources. ● Discuss computing technologies that have changed the world. ● Understand basic cybersecurity threats and safety tips. ● Explore design thinking and digital usage rights.
Lessons	**Research: Effective Keywords (CSS.CC.3-5.6, CSS.RR.3-5.8, CSS.RR.3-5.8.1, CSS.RR.3-5.8.2, CSS.RR.3-5.8.3)** ● Evaluate the effectiveness of different keywords when using a search engine and determine if a source is credible and relevant to their research question. **Impacts of Computing: Introduction (CSS.DA.3-5.9.1)** ● Explain how technology and culture influence each other. **Introduction to Design Thinking (2 classes | CSS.KC.3-5.2, CSS.KC.3-5.2.1, CSS.KC.3-5.2.2, CSS.KC.3-5.2.3, CSS.IDC.3-5.4, CSS.IDC.3-5.4.1, CSS.IDC.3-5.4.2, CSS.GC.3-5.7.2, CSS.GC.3-5.7.4, CSS.GC.3-5.7.5, CSS.RR.3-5.8.3, CSS.DA.3-5.9.2)** ● Use the design thinking process to make a tool accessible to more users. **What Can I Use Online? (CSS.DC.3-5.3.2, CSS.RR.3-5.8, CSS.RR.3-5.8.1, CSS.RR.3-5.8.2)** ● Search for information to answer questions online and provide proper attribution to sources. **Scout's Cybersecurity Adventure: Part 1 (CSS.DC.3-5.3.1)** ● Understand basic cybersecurity concepts, identify common cyber threats, and explain practical tips for staying safe online. **Digital Literacy Quiz (5 questions)**

Unit 10: Optional Interdisciplinary (10 weeks)

In this optional cross-curricular unit, students integrate computer science concepts with academic subjects including math, science, ELA, and social studies. They apply loops, events, conditionals, and variables to build interactive projects that reinforce academic concepts while promoting creativity and problem-solving.

Objectives / Topics Covered	● Use loops and events to model math, science, and storytelling concepts. ● Animate scenes using multiplication, weather patterns, and physical forces. ● Generate sentences using randomization and parts of speech. ● Create interactive programs to explore cultural elements and community adaptation. ● Apply programming skills to demonstrate academic learning across subjects.
Lessons	**Adding with Loops (CSS.CT.3-5.5.2, CSS.CT.3-5.5.4, CSS.CC.3-5.6.2)** ● Use loops to repeat commands and add multi-digit whole numbers based on place value. **Animating Sprites with Multiplication** ● Use multiplication to animate sprites with loops and wait blocks. **Weather and Climate (CSS.CT.3-5.5.4)** ● Use climate data and event blocks to predict and demonstrate typical weather conditions for a specific month. **Balanced and Unbalanced Forces (CSS.CT.3-5.5.4, CSS.CC.3-5.6.2)** ● Describe how balanced and unbalanced forces impact an object's speed and model these forces using conditionals and variables.

	Parts of Speech: Random Sentence Generator (CSS.CT.3-5.5.4) ● Generate random numbers to create simple sentences with nouns, adjectives, verbs, and adverbs. **Creative Storytelling (2 classes | CSS.CT.3-5.5.4)** ● Plan and animate a story using events and sequences. **Communities Adapt to & Modify Their Environment (CSS.CT.3-5.5.4)** ● Use click events to create a scene that shows how communities adapt to or modify their environments. **Choose Your Own Path: Elements of Culture (2 classes | CSS.KC.3-5.2, CSS.DC.3-5.3.2, CSS.CT.3-5.5.2, CSS.CT.3-5.5.3, CSS.CT.3-5.5.4, CSS.CC.3-5.6.1)** ● Identify elements of culture as they create a cultural choose-your-own-path game.

3rd Grade Course Supplemental Materials

Resources	Description
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to their new computer science curriculum.
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 Templates	Empower your students to explore and express their knowledge creatively with our versatile 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**.	