CodeHop

Georgia Computer Science Kindergarten Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Georgia 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. 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.

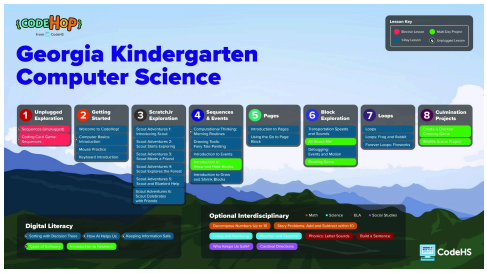
The lessons are delivered in an **"I do, we do, you do"** format, ensuring a gradual release of responsibility and fostering confidence in students as they learn. The concepts taught in this course spiral across grade levels, ensuring that students can build upon their understanding year after year. 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 two-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/26273/overview?lang=en.



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

Course Breakdown

Unit 1: Optional Unplugged Exploration (2 weeks)

In this unplugged unit, students explore the basics of coding without using a device. Through hands-on activities, movement, and teamwork, they learn how to create step-by-step instructions (sequences), laying the foundation for future programming with fun, active experiences.

Objectives / Topics Covered	● Understand sequencing as a fundamental programming concept. ● Create step-by-step instructions to represent a sequence. ● Apply sequences to real-life activities (e.g., choreographing a dance).
Lessons	**Sequences (Unplugged) (CSS.CT.K-2.5, CSS.CT.K-2.5.1, CSS.CT.K-2.5.3, CSS.CT.K-2.5.4)** ● Create a sequence of step-by-step instructions to choreograph a dance routine. **Coding Card Game: Sequences (CSS.CT.K-2.5.3, CSS.CT.K-2.5.4)** ● Work collaboratively to build a sequence of instructions that navigates Scout through a maze.

Unit 2: Getting Started (3 weeks)

In this introductory unit, students are introduced to basic computer operations, including logging in, using input devices like the mouse and keyboard, and understanding the fundamental parts and functions of a computer.

Objectives / Topics Covered	● Log in and navigate the CodeHop Playground. ● Recognize basic computer functions and parts. ● Identify input, output, hardware, and software. ● Practice using the mouse and keyboard for common tasks.
Lessons	**Welcome to CodeHop! (15 minute lesson) (CSS.DA.K-2.9.11)** ● Log in and explore how to use the CodeHop Playground in a short introductory session. **Computer Basics: Introduction (CSS.GC.K-2.7.1, CSS.DA.K-2.9, CSS.DA.K-2.9.2, CSS.DA.K-2.9.3, CSS.DA.K-2.9.7, CSS.DA.K-2.9.8, CSS.DA.K-2.9.11)** ● Learn what a computer is, how we use it, and how to identify and respond to common computer problems; recognize input, output, hardware, and software. **Mouse Practice (CSS.KC.K-2.2.1, CSS.DA.K-2.9.3)** ● Demonstrate mouse control through clicking and dragging in interactive games. **Keyboard Introduction (CSS.KC.K-2.2.1, CSS.KC.K-2.2.2, CSS.KC.K-2.2.3, CSS.DA.K-2.9.3)** ● Practice using letters, numbers, and essential keyboard functions accurately. **Getting Started Quiz (2 questions)**

Unit 3: Programming Exploration (6 weeks)

In this unit, students follow a story-driven sequence of lessons to explore CodeHopJr, gradually learning how to add and modify elements, use motion blocks, and build sequences to animate characters and tell a story.

Objectives / Topics Covered	● Navigate the CodeHopJr interface and add characters and backgrounds. ● Use motion blocks and build sequences to animate characters. ● Create interactive, multi-page stories with events and celebrations.
Lessons	**Scout Adventures 1: Introducing Scout (CSS.EL.K-2.1)** ● Explore the CodeHopJr interface and add characters to the stage.

	Scout Adventures 2: Scout Starts Exploring (CSS.EL.K-2.1) ● Add backgrounds and a new page to create a multi-scene story. **Scout Adventures 3: Scout Meets a Friend (CSS.EL.K-2.1)** ● Delete and edit characters to shape the story. **Scout Adventures 4: Scout Explores the Forest (CSS.EL.K-2.1)** ● Use motion blocks to move characters around the stage. **Scout Adventures 5: Scout and Bluebird Help (CSS.EL.K-2.1)** ● Build a sequence of motion blocks to guide characters in collecting objects. **Scout Adventures 6: Scout Celebrates with Friends (CSS.EL.K-2.1)** ● Create a celebration scene with new characters, pages, backgrounds, and motion sequences using events. **Programming Exploration Quiz (2 questions)**

Unit 4: Sequences & Events (6 weeks)

In this unit, students deepen their understanding of sequencing and events by applying computational thinking to real-life routines, using creative tools, and programming visual effects and interactions.

Objectives / Topics Covered	● Apply sequencing, decomposition, and pattern recognition in everyday tasks. ● Use painting tools to create digital artwork. ● Program character behavior using events, visibility, and size-changing blocks. ● Combine sequences with visual effects to enhance storytelling.
Lessons	**Computational Thinking: Morning Routines (CSS.CT.K-2.5, CSS.CT.K-2.5.3, CSS.CT.K-2.5.4)** ● Use computational thinking to identify patterns, sequence steps, and simplify morning routines. **Drawing Tools: Fairy Tale Painting (CSS.KC.K-2.2, CSS.CC.K-2.6, CSS.CC.K-2.6.1)** ● Use painting tools to design a custom fairy-tale scene. **Introduction to Events (CSS.CT.K-2.5.3, CSS.CT.K-2.5.4, CSS.CT.K-2.5.6, CSS.CC.K-2.6, CSS.CC.K-2.6.1)** ● Create a program that responds to different types of events. **Introduction to Show and Hide Blocks (2 classes | CSS.KC.K-2.2, CSS.CT.K-2.5.3, CSS.CT.K-2.5.4, CSS.CC.K-2.6.1)** ● Use "show" and "hide" blocks in a sequence to control character visibility. **Introduction to Grow and Shrink Blocks (CSS.DA.K-2.9.4)** ● Build a program that changes character size using "grow" and "shrink" blocks. **Sequences & Events Quiz (2 questions)**

Unit 5: Pages (2 weeks)

In this unit, students learn to create multi-page programs and use navigation blocks to move between scenes.

Objectives / Topics Covered	● Create programs with multiple pages. ● Use the “go to page” block to connect scenes and build interactive stories.
Lessons	**Introduction to Pages (CSS.KC.K-2.2, CSS.CT.K-2.5.4, CSS.CC.K-2.6, CSS.CC.K-2.6.1)** ● Create a program that includes multiple pages to expand a story or activity. **Using the Go To Page Block (CSS.KC.K-2.2, CSS.CT.K-2.5, CSS.CT.K-2.5.1, CSS.CT.K-2.5.3, CSS.CT.K-2.5.4)** ● Add a “go to page” block to switch between pages within a program. **Pages Quiz (2 questions)**

Unit 6: Block Exploration (6 weeks)

In this unit, students explore a variety of blocks to create personalized projects, control speed and sound, debug programs, and build simple interactive games.

Objectives / Topics Covered	• Use sound, speed, and speech blocks. • Personalize programs to share information about themselves. • Identify and fix errors in sequences involving events and motion. • Apply coding skills to build an interactive game.
Lessons	**Transportation Speeds and Sounds (CSS.KC.K-2.2, CSS.KC.K-2.2.4, CSS.CT.K-2.5, CSS.CT.K-2.5.1, CSS.CT.K-2.5.3, CSS.CT.K-2.5.4, CSS.CC.K-2.6, CSS.CC.K-2.6.1)** • Use "say" or "sound" blocks and speed blocks to program different modes of transportation. **All About Me! (2 classes | CSS.KC.K-2.2, CSS.CT.K-2.5.1, CSS.CT.K-2.5.3, CSS.CT.K-2.5.4, CSS.CC.K-2.6, CSS.CC.K-2.6.1)** • Create a personalized program that shares favorite things using characters and scenes. **Debugging: Events and Motion (CSS.KC.K-2.2, CSS.CT.K-2.5.1, CSS.CT.K-2.5.2, CSS.CT.K-2.5.5, CSS.CT.K-2.5.6)** • Identify and fix bugs in programs that use events and motion blocks. **Bowling Game (2 classes | CSS.CT.K-2.5.1, CSS.CT.K-2.5.4, CSS.CT.K-2.5.6, CSS.CC.K-2.6.1)** • Apply coding skills to design and program a simple interactive bowling game. **Block Exploration Quiz (2 questions)**

Unit 7: Loops (3 weeks)

In this unit, students explore how loops can be used to repeat actions, allowing for more efficient and dynamic programming.

Objectives / Topics Covered	• Understand and apply loops to repeat sequences in a program. • Use different types of loops, including forever loops, to enhance animations. • Recognize the value of loops in simplifying repetitive code.
Lessons	**Loops (CSS.CT.K-2.5, CSS.CT.K-2.5.1, CSS.CT.K-2.5.2, CSS.CT.K-2.5.3, CSS.CT.K-2.5.4, CSS.CT.K-2.5.6, CSS.CC.K-2.6, CSS.CC.K-2.6.1)** • Create a program using loops and explain how they help repeat code efficiently. **Loops: Frog and Rabbit (CSS.KC.K-2.2, CSS.CT.K-2.5.2, CSS.CT.K-2.5.3, CSS.CC.K-2.6.1)** • Use loops to repeat actions in a program featuring animal characters. **Forever Loops: Fireworks (CSS.KC.K-2.2, CSS.CT.K-2.5.2, CSS.CT.K-2.5.3, CSS.CC.K-2.6.1)** • Create sequences that repeat continuously to animate a fireworks display. **Loops Quiz (2 questions)**

Unit 8: Culmination Projects (4 weeks)

In this unit, students apply their knowledge of events, sequences, and loops to design creative final projects, including a game and an animated scene.

Objectives / Topics Covered	• Program an interactive character with adjustable difficulty. • Combine events, sequences, and loops to build custom animations. • Showcase creativity and coding skills in original projects.
Lessons	**Chicken Crossing Game (2 classes | CSS.KC.K-2.2, CSS.CT.K-2.5.3, CSS.CT.K-2.5.4, CSS.CT.K-2.5.6, CSS.CC.K-2.6.1)** • Program an interactive game with a controllable character and customizable difficulty.

	Wildlife Scene Project (2 classes | CSS.CT.K-2.5, CSS.CT.K-2.5.1, CSS.CT.K-2.5.3, CSS.CT.K-2.5.4, CSS.CT.K-2.5.6, CSS.CC.K-2.6, CSS.CC.K-2.6.1) ● Create a dynamic wildlife scene using events, sequences, and loops.

Unit 9: Digital Literacy (6 weeks)

In this unit, students develop digital literacy by exploring how AI works, learning to protect personal information, choosing appropriate software, and using research skills to find and share information.

Objectives / Topics Covered	● Understand how AI uses data, makes decisions, and helps people. ● Identify private and personal information to stay safe online. ● Compare different types of software for specific tasks. ● Practice research skills and communicate findings through a digital project.
Lessons	**Sorting with Decision Trees (CSS.EL.K-2.1, CSS.CT.K-2.5.2, CSS.DA.K-2.9.1)** ● Explain how AI uses data to make decisions and build a simple decision tree to sort items by rules. **How AI Helps Us (CSS.EL.K-2.1, CSS.IDC.K-2.4.1, CSS.IDC.K-2.4.4, CSS.DA.K-2.9, CSS.DA.K-2.9.1, CSS.DA.K-2.9.5)** ● Learn how AI assistants answer questions and explore how AI supports people in solving problems and transforming jobs. **Keeping Information Safe (CSS.DC.K-2.3, CSS.DC.K-2.3.1, CSS.DC.K-2.3.2, CSS.DC.K-2.3.5)** ● Identify the difference between private and personal information to stay safe online. **Types of Software (CSS.GC.K-2.7.1)** ● Compare software applications and choose the right one for different computing tasks. **Introduction to Research (2 classes | CSS.DC.K-2.3.7, CSS.DC.K-2.3.8, CSS.CC.K-2.6.2, CSS.CC.K-2.6.3, CSS.RR.K-2.8, CSS.RR.K-2.8.1, CSS.RR.K-2.8.2, CSS.RR.K-2.8.3, CSS.RR.K-2.8.4, CSS.RR.K-2.8.6, CSS.RR.K-2.8.7)** ● Find information using research tools and create a visual program to share findings. **Digital Literacy Quiz (2 questions)**

Unit 10: Optional Interdisciplinary (8 weeks)

In this optional unit, students integrate computer science with core subjects—math, science, ELA, and social studies—by creating interactive projects that reinforce academic concepts through coding.

Objectives / Topics Covered	● Apply coding concepts like events, sequences, and audio to support math, literacy, science, and social studies. ● Create interactive programs that reinforce classroom learning in multiple subjects. ● Visually represent academic concepts such as number sense, phonics, and weather patterns.
Lessons	**Decompose Numbers Up to 10** ● Create an interactive program using events to show how numbers can be broken apart in different ways. **Story Problems: Add and Subtract within 10** ● Build a scene that illustrates an addition or subtraction word problem. **Living and Nonliving** ● Classify items as living or nonliving and use tap events to trigger character actions. **Weather and Seasons** ● Program a weather animation that shows changes across different seasons using sequences.

	Phonics: Letter Sounds ● Create a phonics game that uses the “on tap” event and recorded audio for letter sounds. **Build a Sentence** ● Develop an interactive program that uses events to build and read sentences aloud. **Who Keeps Us Safe?** ● Create a project that explains the role of an authority figure in the community. **Cardinal Directions** ● Use events to create an interactive compass rose demonstrating north, south, east, and west.

Kindergarten Course Supplemental Materials

Resources	**Description**
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to computer science with CodeHop.
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.
These resources and more are found on the **Elementary Resources Page**.	