CodeHS

Florida Computer Science Foundations Kindergarten Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Florida Computer Science Foundations Kindergarten Course** introduces students to foundational programming concepts through **ScratchJr**, 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 **contact hours**, with each lesson approximately 30 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 **ScratchJr** embedded and saved in students' accounts. 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/FL_K/overview

Course Breakdown

Optional Unplugged Exploration

In this unplugged unit, students develop foundational programming skills such as sequencing, pattern recognition, and algorithmic thinking through hands-on activities that encourage collaboration, problem-solving, and computational reasoning.

Objectives / Topics Covered	• Create simple sequences. • Practice giving and following directions. • Recognize patterns and develop problem-solving skills.

Lessons	**Sequences (Unplugged)** ● Create a step-by-step sequence of dance moves to understand how order matters in instructions. **Coding Card Game: Sequences** ● Work together to give clear directions that move a character through a maze, practicing how to build a sequence of instructions. **Coding Card Game: Sequences 2** ● Build on the previous activity by creating new sequences and solving different mazes as a team. **Acting with Events** ● Act out simple sequences where something happens to cause an action, introducing the idea of events in a fun and physical way.

Unit 1: Getting Started (4 weeks)

In this introductory unit, students will begin developing basic computer skills and explore foundational concepts in computer science. Through hands-on practice and familiar routines, they'll learn how computers work, how to use a keyboard, and how to think like a computer scientist by recognizing patterns and sequencing steps.

Objectives / Topics Covered	● Log in and navigate the Playground. ● Identify and explain basic computer parts and their functions. ● Develop responsible digital citizenship habits. ● Practice using a keyboard. ● Apply computational thinking to everyday routines.
Lessons	**Welcome to CodeHS! (15 minute lesson)** ● Introductory lesson to help students log in and explore the Playground; perfect as a warm-up or standalone activity. **Keyboard Introduction** ● Explore the keyboard by identifying and using letters, numbers, and simple function keys. **Computer Basics: Introduction** ● Learn what a computer is, how it's used, and how input, output, hardware, and software work together. **Introduction to Responsible Technology Use** ● Identify ways to use technology safely and responsibly. **Computational Thinking: Morning Routines** ● Break down a morning routine into steps and recognize patterns using computational thinking strategies.

Unit 2: ScratchJr Exploration (6 weeks)

In this story-based unit, students will explore the basics of block-based programming in ScratchJr, learning how to animate characters, build simple stories, and begin thinking like programmers through experimentation and play.

Objectives / Topics Covered	● Learn to navigate the ScratchJr interface. ● Add, delete, and modify characters and backgrounds. ● Create sequences. ● Use event blocks to trigger movement and interactions.
Lessons	**Scout Adventures 1: Introducing Scout** ● Explore the ScratchJr interface and add characters to a project. **Scout Adventures 2: Scout Starts Exploring** ● Add backgrounds and a new page to a ScratchJr project. **Scout Adventures 3: Scout Meets a Friend** ● Delete and modify characters.

	Scout Adventures 4: Scout Explores the Forest ● Use motion blocks to move characters around the page. **Scout Adventures 5: Scout and Bluebird Help** ● Build a sequence of motion blocks to move characters and collect objects. **Scout Adventures 6: Scout Celebrates with Friends** ● Create a celebration scene by adding characters, pages, backgrounds, and sequences of motion blocks with events.

Unit 3: Sequences and Events (6 weeks)

In this unit, students will dive deeper into computer science by creating precise sequences and using events to control actions in a program.

Objectives / Topics Covered	● Create sequences using motion and event blocks. ● Use events to trigger animations and interactions. ● Make characters appear, disappear, grow, and shrink.
Lessons	**Introduction to Events** ● Create a program using different types of events. **Sequences: Snowball Fight** ● Create a program using multiple sequences. **Careers in CS: Litter Free Communities** ● Explain how computer science can help solve community problems by creating a program to collect and sort litter. **Introduction to Grow and Shrink Blocks** ● Create a program that uses "grow" and "shrink" blocks to change the size of characters. **Finding an Unknown in a Word Problem** ● Use events and sequences to create a number story for addition and subtraction. **Place Value: Ones and Tens** ● Use animation to model groups of ones, fives, and tens.

Unit 4: Pages (5 weeks)

In this unit, students will use Pages in ScratchJr to organize ideas across multiple scenes, learning how to structure digital stories, plan projects, and develop computational thinking skills.

Objectives / Topics Covered	● Create multi-page ScratchJr programs. ● Use the “go to page” block to switch between pages.
Lessons	**Introduction to Pages** ● Create a program with multiple pages. **Using the Go To Page Block** ● Build a program that switches between pages using the “go to page” block for interactive navigation. **Decompose Numbers Up to 10** ● Use events to visualize algebraic thinking and decomposing numbers. **Story Problems: Add and Subtract within 10** ● Create a scene that represents an addition or subtraction story problem. **Greater Than and Less Than: Single-Digit Numbers** ● Create a program that compares the values of numbers using events.

Unit 5: Block Exploration (4 weeks)

In this unit, students will investigate how different types of programming blocks work together in ScratchJr, deepening their understanding of commands, control, motion, and how to combine them to create engaging projects.

Objectives / Topics Covered	● Use speed, sound, and say blocks. ● Identify and fix errors in code.
Lessons	**Introduction to Speed Blocks** ● Create a program that uses different speed blocks to animate characters. **Animal Sounds** ● Use "say" or "sound" blocks to program a character to speak. **Debugging: Events and Motion** ● Find and fix bugs in a program that uses event and motion blocks. **Creating Shapes** ● Program shapes that can respond to user interaction.

Unit 6: Loops (3 weeks)

In this unit, students will explore the concept of loops, learning how to use repetition in programs to make their code more efficient and solve problems more creatively.

Objectives / Topics Covered	● Use loops to simplify sequences. ● Recognize patterns that can be repeated in code.
Lessons	**Loops** ● Create a program using loops and explain how loops are used to repeat code. **Forever Loops: Fireworks** ● Create sequences that repeat continuously while the program is running, using forever loops to animate ongoing actions. **Patterns and Music** ● Create musical patterns using sequences and loops.

Unit 7: Culmination Project (2 weeks)

In this unit, students will apply what they've learned by designing and creating their own ScratchJr animation showcasing their programming skills, creativity, and understanding of computer science concepts.

Objectives / Topics Covered	● Use sequences, events, and loops in an original project.
Lessons	**All About Me! (2 part lesson)** ● Create a program that tells information about a student's favorite things.

Unit 8: Digital Literacy (6 weeks)

In this unit, students will learn the basics of digital citizenship, including how to navigate technology safely, make responsible choices online, and understand the roles computers play in their lives and communities.

Objectives / Topics Covered	● Use technology safely and responsibly. ● Explore the role of computers and the internet in everyday life. ● Collect, organize, and analyze data. ● Recognize examples of artificial intelligence in everyday tools.
Lessons	**Keeping Information Safe** ● Identify the difference between personal and private information and learn how to keep information safe. **What Can Data Tell Us?** ● Collect, organize, and analyze data about school transportation.

	What Can AI Do? ● Identify tools that use AI and compare tasks that are better suited for humans vs AI. **Using Networks to Connect** ● Learn how people and devices connect and share information using networks. **Introduction to Research (2 part lesson)** ● Use simple research tools to find information and create a project that shares findings visually.

Interdisciplinary Connections (Supplemental)

In this unit, students strengthen their programming skills by applying them to interdisciplinary concepts in math, science, social studies, and ELA. These flexible, supplemental lessons can be integrated throughout the year to enrich core instruction and provide meaningful, real-world connections across subjects.

Objectives / Topics Covered	● Develop cross-curricular projects using events and sequences. ● Integrate computer science with core content areas to reinforce real-world applications.
Lessons	**Living and Nonliving** ● Classify items as living or nonliving using “on tap” events. **Weather and Seasons** ● Use sequences to program a weather animation. **How Living Things Survive** ● Illustrate how living things survive in their environment. **Comparing Organisms** ● Create a program that groups animals by similar characteristics. **Types of Motion** ● Use events to program an animation displaying types of motion. **Needs of Plants** ● Use events and messages to model the needs of plants. **Build a Sentence** ● Use events to write sentences and then read them aloud. **Storytelling Animations** ● Create a program that retells the details of a story in the correct order. **Original Story Animations - Personal Story (2 part lesson)** ● Design a program to animate an original story and describe their program development process. **Who Keeps Us Safe?** ● Create a program that describes the role of an authority figure. **Cardinal Directions** ● Use events to create an interactive compass rose.

Florida Computer Science Kindergarten 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.
All of these resources and more are found on the **Elementary Resources Page**.	