CodeHS

Florida Computer Science Foundations 4th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Florida Computer Science Foundations 4th 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 **contact hours**, each 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 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_4/overview

Course Breakdown

Optional Review

In this optional unit, students are introduced to key computer science vocabulary, practice logging into and navigating the Playground, and explore basic programming in Scratch. They also learn to use the coordinate plane to position sprites, building spatial reasoning and foundational coding skills.

Objectives / Topics Covered	• Log in and navigate the Playground. • Understand and use key computer science vocabulary. • Create a simple Scratch program to demonstrate basic programming skills. • Use the coordinate plane to position sprites.
Lessons	**Welcome to CodeHS!** • Learn how to log in and explore the Playground to prepare for future lessons.

	Introduction to Computer Science and Scratch ● Define key computer science vocabulary and create a simple Scratch program to apply foundational programming concepts. **The Coordinate Plane** ● Use the coordinate plane in Scratch to position sprites and create an open-ended animation.

Unit 1: Getting Started (3 weeks)

In this unit, students explore how their online actions can promote positive behavior and develop a code of conduct for responsible internet use. They also apply computational thinking skills to design a school, using strategies like breaking down tasks, identifying patterns, and organizing ideas logically.

Objectives / Topics Covered	● Explore how online actions can promote positivity. ● Apply computational thinking skills. ● Learn about the components and functions of computing systems.
Lessons	**Internet Positivity** ● Explore how actions can promote positivity online and create a class code of conduct for responsible internet use. **Exploring Computing Systems** ● Explore the components and functions of computing systems and learn basic troubleshooting techniques. **Computational Thinking: Design a School** ● Apply computational thinking by breaking down tasks, identifying patterns, and organizing ideas to design a school layout.

Unit 2: Sequences and Events (8 weeks)

In this unit, students deepen their understanding of events by programming with multiple event types and using broadcast messages to coordinate sprite interactions. They also design algorithms, collaborate through pair programming, and apply coding skills creatively to build interactive projects.

Objectives / Topics Covered	● Create programs using multiple types of event blocks to control when actions happen. ● Develop and compare algorithms to solve problems efficiently. ● Use broadcast messages to coordinate interactions between sprites.
Lessons	**Creating Algorithms** ● Write and compare different algorithms to determine which one best fits the goal of the program. **Pair Programming: Create a Band (2 part lesson)** ● Work with a partner to design and code a musical band in Scratch using keyboard inputs to control sprites. **Careers in CS: Health and Fitness** ● Create an animation to demonstrate how technology can be used in health and fitness. **Exploring the Water Cycle** ● Explain the phases of the water cycle using broadcast messages. **Classifying Shapes Using Lines and Angles** ● Categorize shapes based on the properties of their lines and angles. **Choose Your Own Path: Elements of Culture (2 part lesson)** ● Create a choose-your-own-path game that explores elements of culture through branching storylines and interactive programming.

Unit 3: Loops (3 weeks)

In this unit, students learn to use different types of loops and understand how repetition can simplify code. They also practice debugging by breaking down and analyzing a program to identify and fix errors, strengthening their problem-solving skills.

Objectives / Topics Covered	● Use different types of loops to simplify algorithms. ● Break down and analyze code to identify and fix errors.
Lessons	**Loops: Catch the Ball** ● Use two types of loops in Scratch to create a simple interactive game where players catch a moving ball. **Debugging: Mazes** ● Break down and analyze a Scratch program to find and fix errors, focusing on loops and logical flow. **Animating Sprites with Factors** ● Use factors to animate sprites with loops and wait blocks.

Unit 4: Conditionals and Variables (5 weeks)

In this unit, students build interactive programs using conditionals and variables to make decisions, track information, and enhance gameplay.

Objectives / Topics Covered	● Use if/then conditionals to control decision-making in programs. ● Create and update variables to store and track information.
Lessons	**Scout's Quest: Conditionals** ● Use if/then conditionals to build a program that makes decisions based on input or events. **Division and Conditionals** ● Solve division problems using conditionals. **Scout's Quest: Variables** ● Create and use variables to track information such as points in a game. **AI Chatbots** ● Describe how AI representations support reasoning by training a model AI chatbot program. **Sound Frequency and Amplitude** ● Use variables to control the rate and strength of the sound in a speaker sprite.

Unit 5: Clones and Functions (5 weeks)

In this unit, students explore how to use clones and functions to create more efficient and dynamic programs. They build animations, design custom functions with inputs—including booleans and numbers—and apply these concepts creatively.

Objectives / Topics Covered	● Explore how clones affect program behavior. ● Use functions with number and boolean inputs to trigger actions based on conditions.
Lessons	**Introduction to Clones** ● Create an animation using clones in Scratch and explore how clone behavior affects the program's outcome. **Scout's Quest: Functions with Boolean Inputs** ● Build a function that uses a boolean input to perform different actions, such as checking if a password is correct.

	Scout's Quest: Functions with Number Inputs ● Use number inputs in a function to create a drawing that changes based on input values. **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.

Unit 6: Culmination Projects (4 weeks)

In this unit, students apply their understanding of conditionals, variables, booleans, and events to design and build interactive games and programs.

Objectives / Topics Covered	● Apply key programming concepts including conditionals, variables, and events. ● Demonstrate mastery of coding skills through creative problem-solving and game design.
Lessons	**Click-a-Mole (2 part lesson)** ● Create an interactive Whack-a-Mole style game using conditionals, variables, and events to demonstrate mastery of core programming concepts. **Designing Solutions for Accessibility (2 part lesson)** ● Explore how to apply design thinking to create digital tools and experiences that are accessible to everyone.

Unit 7: Digital Literacy (8 weeks)

In this unit, students develop digital literacy by exploring safe online behaviors, understanding how information travels across the internet, and creating interactive projects to communicate what they've learned. They investigate technology's impact on culture, follow the inquiry process to present data, and begin to explore concepts such as digital ownership, accessibility, and cybersecurity.

Objectives / Topics Covered	● Practice safe online behavior. ● Follow the inquiry process to collect, analyze, and present data visually. ● Learn how information travels through the Internet.
Lessons	**Our Words Have Power (Cyberbullying)** ● Explain what cyberbullying is and how it affects others. **Scout's Cybersecurity Adventure: Part 2** ● Practice safe online habits and learn about tools and technologies that help protect personal information. **Networks, Packets, and the Internet** ● Learn how information travels across the internet by modeling how messages are broken into packets, sent, and reassembled, and create a secure classroom communication system. **Give Credit When You Use It** ● Research information online to answer questions and give credit to sources properly. **Inquiry Project: Data Bar Graph (2 part lesson)** ● Follow the inquiry process to gather data and modify a Scratch program to display findings as a bar graph. **Humans Vs. Machines** ● Compare and contrast human and computer performance on similar tasks. **Ethical and Responsible Use of Generative AI** ● Describe the pros and cons of generative AU and complete a class Code of Conduct to follow when using AI.

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	● Use conditionals and loops to build simulations and practice content-area skills. ● Create animations to connect programming to the real world.
Lessons	**Multi-Digit Multiplication and Conditionals** ● Use conditionals to review multiplication with multi-digit factors. **Multiplying Fractions by Whole Numbers** ● Create an interactive program to demonstrate multiplying fractions by whole numbers. **Naming Numbers Game** ● Use variables and operators to read and write numbers in number and expanded form. **Exploring Heat** ● Use events to communicate how heat energy from the sun affects objects on earth. **Sensing and Responding to the Environment** ● Model how animals use their senses to respond to their environment. **Plant and Animal Cells** ● Use broadcast messages to create an interactive program about plant and animal cells. **How We See: Light Reflection** ● Use conditionals to demonstrate how light reflects from objects and enters the eye to allow objects to be seen. **Wave Generator** ● Use variables and loops to draw wave patterns. **Grammar Quiz Game** ● Use conditionals to create a quiz game. **Mad Libs Project (2 part lesson)** ● Use lists to create a mad libs game. **Rights and Responsibilities** ● Use variables and events to create a voting program.

Florida Computer Science 4th 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 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 **Elementary Resources Page**.	