CodeHS

Florida Computer Science Foundations 2nd Grade Course Syllabus
One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Florida Computer Science Foundations 2nd Grade 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 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 **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_2/overview

Course Breakdown

Optional Review

This optional review unit is designed to support students who need more time exploring ScratchJr or who would benefit from additional practice before or after completing core lessons. Students will reinforce key programming concepts including navigating ScratchJr, using event and message blocks to trigger actions, and applying repeat and forever loops to build interactive and animated programs.

Objectives / Topics Covered	● Log in and navigate the Playground. ● Reinforce understanding of basic programming concepts such as events, loops, and message blocks.

Lessons	**Welcome to CodeHS!** ● Learn how to log in and use the Playground; a quick intro lesson to get started with the platform. **Events** ● Learn what an event is in programming and use event blocks to trigger actions in a project. **Introduction to Repeat Loops** ● Use repeat loops to run a section of code multiple times and simplify repeated actions. **Forever Loop Dance Party** ● Create a fun animation where characters repeat movements continuously using the "repeat forever" loop. **Introduction to Message Events** ● Build a relay-style animation where characters interact using message blocks to control the flow of the program.

Unit 1: Getting Started (3 weeks)

In this unit, students will explore the basics of computing by learning the functions of computers and their components, practice identifying positive and negative online behaviors, and apply foundational computational thinking skills like pattern recognition, sequencing, and task decomposition to real-life routines.

Objectives / Topics Covered	● Understand the basic components of a computer. ● Learn responsible ways to use technology. ● Apply computational thinking in everyday routines.
Lessons	**Computer Basics: Connections** ● Learn what a computer is, how we use it, and what to do when it doesn't work. **Practicing Responsible Technology Use** ● Explore ways to use technology safely, respectfully, and responsibly in daily life. **Computational Thinking: School Day Routines** ● Use computational thinking to find patterns, break down routines, and organize steps to better understand everyday school activities.

Unit 2: Sequences and Events (8 weeks)

In this unit, students build foundational programming skills by creating and adjusting algorithms, using the grid to control movement, and designing interactive programs with events.

Objectives / Topics Covered	● Develop simple algorithms to control character movement and behavior. ● Use events to trigger actions. ● Apply grid-based positioning to guide character actions.
Lessons	**Debugging: Events and Sequences** ● Practice finding and fixing errors in code to make a program work as expected. **Introduction to the Grid** ● Use the grid feature in ScratchJr to move characters accurately to specific locations on the page. **Careers in CS: Coding for Fashion-Retail** ● Explain how coding helps create and improve fashion designs by programming a fashion character. **Algorithms: Connecting a Path** ● Create and adjust simple algorithms that guide characters based on their starting position, size, and shape. **Data Patterns and Predictions** ● Identify patterns in data visualizations and use events in a program to show predictions based on the data.

	Events and Money ● Create a program that tells the value of a set of coins. **Create Your Own Story Problem** ● Use events to create a scene that helps to solve addition and subtraction word problems. **Identify Shapes by Attributes** ● Create a program to draw and identify shapes with specified attributes.

Unit 3: Message Events (5 weeks)

Students will deepen their understanding of program control by using message events to trigger actions, model cycles, and coordinate activity across multiple pages.

Objectives / Topics Covered	● Use message events to control the flow of a program. ● Navigate between pages in a program.
Lessons	**Message Events: Scout Plays in the Forest** ● Use message events to control when and how different parts of a program run. **Programming a Cycle** ● Model a real-world cycle by using message blocks to create repeating sequences in a program. **Pages: Scout's Travels** ● Use message events to move between pages in a multi-scene ScratchJr program. **Exploring Computer Networks** ● Understand how networks connect devices and model how messages are sent and received using message blocks. **3D Shapes** ● Draw 3D shapes and create a quiz game to review their attributes.

Unit 4: Loops (5 weeks)

Students will apply loops to identify patterns, build timers, and simplify repeated actions in code. They also strengthen their debugging skills by correcting errors involving loops and message events, and demonstrate creativity by developing and animating an original story.

Objectives / Topics Covered	● Use loops to simplify repeated actions in a program. ● Debug programs by finding and fixing errors.
Lessons	**Loops: Follow the Path** ● Identify patterns and use loops to create a program that repeats actions along a path. **Debugging: Message Events and Loops** ● Find and fix errors in programs that use both loops and message events. **Original Story Animations – Fiction Story (2 part lesson)** ● Develop and animate an original fiction story using programming concepts like sequencing and loops. **Seed Dispersal** ● Use message events and loops to model how an animal can help disperse seeds.

Unit 5: Culmination Projects (7 weeks)

In this unit, students apply their full range of computer science skills to create interactive games and animations using sequences, events, loops, messages, and pages.

Objectives / Topics Covered	● Design interactive games that incorporate events, loops, messages, and sequences. ● Demonstrate mastery of computer science concepts through original projects.

Lessons	**Code Block Review (2 part lesson)** ● Use a variety of coding blocks in a program and explain what each one does. **Racing Game (2 part lesson)** ● Create an interactive racing game using events, loops, and message blocks. **Design an Adventure Game (3 part lesson)** ● Design a multi-page, story-based adventure game that showcases skills learned throughout the course.

Unit 6: Digital Literacy (8 weeks)

Students will learn how to protect personal information with secure login practices, collect information and analyze data, and understand how AI learns and makes decisions.

Objectives / Topics Covered	● Understand the importance of secure login information. ● Collect, analyze, and visually present data. ● Conduct research using reliable sources and share findings visually.
Lessons	**Password Protectors** ● Learn why usernames and passwords are important and practice ways to keep login information safe. **Sorting with Decision Trees** ● Explain how AI uses data to learn and make decisions by creating a simple decision tree to sort items based on rules. **Machine Learning: AutoDraw** ● Describe how AutoDraw uses AI to recognize and suggest drawings. **Advanced Data and Programming (3 part lesson)** ● Ask a question, collect data through a survey, and create a program to present the results visually. **Choice Research (2 part lesson)** ● Research a topic using trusted sources and create a visual program to share findings.

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 events, loops, and sequences to build simulations and practice content-area skills. ● Create animations to connect programming to the real world.
Lessons	**Story Problems: Add and Subtract within 100** ● Model a math story problem in a program. **Counting with Mazes** ● Design a maze requiring a specific number of steps. **Place Value: Ones, Tens, and Hundreds** ● Create an interactive program to connect a digit's place in a number to its value. **Algebraic Thinking: Finding a Two-Digit Unknown** ● Create a program to solve for unknown numbers in number stories. **Greater Than and Less Than: Three-Digit Numbers** ● Use events to compare three-digit numbers. **Designing Solutions from Nature** ● Use events to show how humans survive in their environments. **Changing Landforms** ● Create a sequence to show how a volcanic eruption changes Earth's surface. **Properties of Matter** ● Program a robot to make accurate decisions and group different types of matter.

	Changes in the Environment ● Create an animation to model environmental changes. **Preventing Erosion** ● Create a program to compare multiple solutions designed to slow or prevent wind and water from changing the shape of the land. **Communities Modify Their Environment** ● Create a program that shows how people modify their environment.

Optional Preparing for Next Year

In this optional unit, students expand their programming knowledge by using conditionals in unplugged activities and transitioning from ScratchJr to Scratch. They gain experience with the Scratch interface and apply familiar concepts like events and loops to create simple programs in a new coding environment.

Objectives / Topics Covered	● Use conditionals to guide character action. ● Transition from ScratchJr to Scratch. ● Apply familiar programming concepts, such as events and loops, in a new coding environment.
Lessons	**Coding Card Game: Conditionals** ● Work together to create a sequence of instructions using conditionals to guide a character through a maze. **Coding Card Game: Conditionals 2** ● Continue practicing conditionals by solving new maze challenges with team-based instruction sequences. **From ScratchJr to Scratch** ● Explore the basic interface of the Scratch editor and create a simple program using familiar concepts. **ScratchJr to Scratch: Events and Loops** ● Create a program in Scratch that uses both events and loops to animate actions and build interactivity.

Florida Computer Science 2nd 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.
All of these resources and more are found on the **Elementary Resources Page**.	