



Florida Foundations of Computer Science: 2nd Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Florida Foundations of Computer Science: 2nd Grade 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. This course emphasizes creativity and collaboration, providing students with a solid base in computer science concepts and digital literacy.

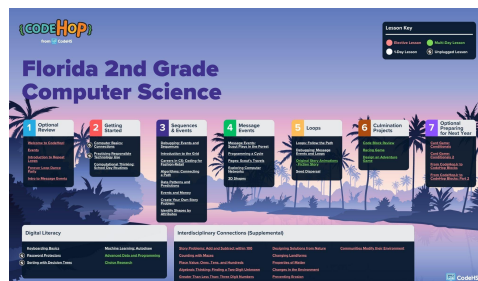
Learning Environment: This course is teacher-led and includes ready-to-use lessons following a consistent structure: Introduction, Guided Practice, Independent Practice, Extension, and Reflection. Instruction follows an “I do, we do, you do” model and incorporates spiral review to reinforce concepts and build confidence over time.

The course includes 36 lessons, each approximately 45 minutes long, providing a full year of instruction when taught once per week. While the course allows for instructional flexibility, some lessons are required to fully meet state computer science standards and are clearly identified within the syllabus. All Digital Literacy lessons are required to ensure full standards alignment, as they address essential non-programming computer science concepts. Required lessons are labeled with the specific standards they address to support planning and compliance.

Programming Environment: Students will write and run programs that are 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



A clickable PDF can be found at <https://codehs.com/FL-K-5Roadmaps>

Course Breakdown

Optional Review

These optional lessons review key first grade concepts—events, loops, and message events—before students begin second grade content.

Objectives / Topics Covered	• Log in and navigate CodeHopJr • Use event blocks in a program • Repeat code with loops • Trigger interactions with message events
Lessons	Welcome to CodeHop! • Log in and explore the CodeHopJr Playground. Events • Explain what an event is and use multiple event blocks in a program. Introduction to Repeat Loops • Use repeat loops to run a section of code multiple times. Forever Loop Dance Party • Use a repeat-forever loop to make characters repeat actions. Introduction to Message Events • Program a relay race using messages to make characters interact.

Unit 1: Getting Started (3 lessons)

Students build foundational understanding of how computers work and responsible technology use, and apply computational thinking to school routines.

Objectives / Topics Covered	• Identify hardware, software, input, and output • Explain how computer parts work together • Use technology safely and responsibly • Apply computational thinking to routines
Lessons	Computer Basics: Connections (SC.2.CC.1.1, SC.2.CO.1.1, SC.2.TI.1.1) • Identify computer parts and explain how input, output, hardware, and software work together. Practicing Responsible Technology Use (SC.2.CO.1.2, SC.2.HS.2.1, SC.2.HS.2.2) • Demonstrate ways to use technology safely and responsibly. Computational Thinking: School Day Routines (SC.K12.CTR.3.1, SC.K12.CTR.4.1, SC.K12.CTR.5.1) • Break down and sequence school-day routines using computational thinking.

Unit 2: Sequences and Events (8 lessons)

Students program with sequences, events, the grid, and algorithms while exploring debugging, careers, data, money, and geometry.

Objectives / Topics Covered	• Find and fix bugs in code • Use the grid and algorithms to move characters • Create event-driven programs • Apply CS to data, money, and shapes
Lessons	Debugging: Events and Sequences (SC.K12.CTR.2.1, SC.K12.CTR.1.1, SC.2.PE.1.1) • Find and fix errors in provided code. Introduction to the Grid (SC.K12.CTR.2.1, MA.K12.MTR.1.1, SC.2.PE.1.1) • Use the grid to move characters to a specific location on the stage.

	Careers in CS: Fashion-Retail (SC.2.TI.1.1, SC.2.TI.1.2, SC.2.PE.1.1) - Explain how coding supports fashion design and animate a fashion character. Algorithms: Connecting a Path - Create and adjust algorithms to move characters by size, shape, and position. Data Patterns and Predictions (SC.2.CO.1.5) - Identify patterns in data and program events to communicate trends and predictions. Events and Money (MA.K12.MTR.1.1, MA.K12.MTR.3.1, MA.K12.MTR.4.1) - Create a program that tells the value of a set of coins. Create Your Own Story Problem (SC.2.PE.3.2, SC.K12.CTR.2.1, SC.2.PE.1.1) - Use events to build a scene that solves addition and subtraction word problems. Identify Shapes by Attributes (MA.K12.MTR.1.1, MA.K12.MTR.3.1, MA.K12.MTR.4.1) - Draw and identify shapes with specified attributes.
--	--

Unit 3: Message Events (5 lessons)

Students use message events to control program flow, model cycles, navigate pages, and explore networks and 3D shapes.

Objectives / Topics Covered	- Control program flow with message events - Model cycles with messages - Navigate pages using messages - Model networks and 3D shapes
Lessons	Message Events: Scout Plays in the Forest (SC.K12.CTR.2.1, SC.2.PE.1.1) - Use message events to control the flow of a program. Programming a Cycle (SC.2.PE.3.3, SC.K12.CTR.2.1, SC.2.PE.1.1) - Use message events to model a cycle. Pages: Scout's Travels - Use messages to move Scout between pages in a program. Exploring Computer Networks (SC.2.CO.1.8, ELA.K12.EE.6.1, SC.2.CO.1.9) - Model how networks connect devices and share information using message blocks. 3D Shapes (MA.K12.MTR.1.1, MA.K12.MTR.3.1, MA.K12.MTR.4.1) - Draw 3D shapes and create a quiz game about their attributes.

Unit 4: Loops (5 lessons)

Students use loops to create programs and debug code, applying them to storytelling and science.

Objectives / Topics Covered	- Identify patterns and program with loops - Debug message event and loop errors - Animate an original story - Model science concepts with loops and messages
Lessons	Loops: Follow the Path (SC.2.PE.3.1) - Identify patterns and create a program using loops. Debugging: Message Events and Loops (SC.K12.CTR.2.1, SC.2.PE.3.1, SC.2.PE.1.1) - Find and fix message event and loop errors in provided code. Original Story Animations - Fiction Story (2 classes SC.2.CO.1.5, SC.2.CO.1.3, SC.2.PE.3.1) - Develop and animate an original fiction story. Seed Dispersal (SC.2.CC.1.4, SC.2.PE.3.1, SC.2.PE.1.1) - Use message events and loops to model how animals help disperse seeds.

Unit 5: Culmination Projects (7 lessons)

Students combine events, loops, and messages to build reviews, games, and a multi-page adventure.

Objectives / Topics	- Use a variety of coding blocks and explain their function - Combine events, loops, and messages
------------------------	--

Covered	• Build interactive games • Create a multi-page adventure game
Lessons	Code Block Review (2 classes SC.K12.CTR.3.1, SC.2.PE.3.1, SC.2.PE.1.1) • Use a variety of coding blocks and explain their function within the program. Racing Game (2 classes SC.2.PE.3.1, SC.K12.CTR.1.1, SC.2.PE.1.1) • Create an interactive racing game with events, loops, and messages. Design an Adventure Game (3 classes SC.K12.CTR.2.1, SC.2.PE.3.1, SC.2.PE.1.1) • Build a story-based, multi-page game using learned computer science skills.

Unit 6: Digital Literacy (8 lessons)

Students build digital citizenship and literacy skills including typing, password safety, artificial intelligence, machine learning, data, and research.

Objectives / Topics Covered	• Type simple words and sentences • Protect usernames and passwords • Explain how AI learns and sorts data • Collect and present data and research
Lessons	Keyboarding Basics (SC.2.CO.1.3) • Type simple words and sentences on a computer. Password Protectors (SC.2.HS.1.2, SC.2.HS.1.3, SC.2.HS.1.1) • Explain the importance of usernames and passwords and keep login information safe. Sorting with Decision Trees (SC.2.PE.2.1, SC.2.PE.2.2, SC.2.PE.2.3) • Explain how AI uses data and build a decision tree to sort items by rules. Machine Learning: AutoDraw (SC.2.TI.1.3) • Describe how AutoDraw uses AI and a classifier to recognize and suggest drawings. Advanced Data and Programming (3 classes SC.2.CC.1.1, SC.2.PE.2.1, SC.2.PE.2.2) • Develop a question, collect survey data, and program a visual presentation. Choice Research (2 classes SC.2.CC.1.1, SC.2.CC.1.3, SC.2.CO.1.3) • Collect and assess sources to answer a research question and share findings visually.

Interdisciplinary Connections (Supplemental)

These optional lessons connect computer science to math, science, and social studies through interactive programs.

Objectives / Topics Covered	• Apply programming across subject areas • Model math concepts like place value and comparison • Model earth and environmental science • Use events and conditionals in context
Lessons	Story Problems: Add and Subtract within 100 (SC.2.PE.3.2, SC.K12.CTR.3.1, SC.K12.CTR.5.1) • Program a model of a math story problem. Counting with Mazes (SC.K12.CTR.2.1, MA.K12.MTR.3.1, SC.2.PE.1.1) • Use counting skills to design a maze requiring a specific number of steps. Place Value: Ones, Tens, and Hundreds (MA.K12.MTR.1.1, MA.K12.MTR.3.1, MA.K12.MTR.4.1) • Program an interactive model connecting a digit's place to its value. Algebraic Thinking: Finding a Two-Digit Unknown (MA.K12.MTR.1.1, MA.K12.MTR.2.1, MA.K12.MTR.3.1) • Build an interactive program to solve for large unknown numbers in number stories. Greater Than and Less Than: Three-Digit Numbers (MA.K12.MTR.1.1, MA.K12.MTR.3.1, MA.K12.MTR.4.1) • Use events to compare three-digit numbers. Designing Solutions from Nature (SC.K12.CTR.2.1, SC.K12.CTR.7.1, SC.2.PE.1.1) • Program how humans mimic plants and animals to survive in their environment.

	Changing Landforms (SC.2.CC.1.4, ELD.K12.ELL.SC.1, SC.K12.CTR.7.1) - Sequence how a volcanic eruption changes Earth's surface. Properties of Matter (SC.K12.CTR.2.1, SC.K12.CTR.7.1, SC.2.PE.1.1) - Use conditionals to group different types of matter by their properties. Changes in the Environment (SC.K12.CTR.2.1, SC.K12.CTR.3.1, SC.2.PE.1.1) - Animate environmental changes and identify their causes. Preventing Erosion (SC.K12.CTR.2.1, SC.K12.CTR.7.1, SC.2.PE.1.1) - Compare solutions designed to slow or prevent erosion. Communities Modify Their Environment (SC.K12.CTR.3.1, SC.K12.CTR.5.1) - Program how people modify their environment in a community.
--	--

Preparing for Next Year (4 lessons)

These forward-looking lessons introduce conditionals and transition students from CodeHopJr to the CodeHop block editor.

Objectives / Topics Covered	- Explore conditionals unplugged - Navigate the CodeHop editor - Create programs with events and loops in CodeHop
Lessons	Coding Card Game: Conditionals (SC.2.PE.1.1, SC.2.PE.3.3, ELA.K12.EE.4.1) - Work as a team to build conditional instructions to move Scout through a maze. Coding Card Game: Conditionals 2 (SC.2.PE.1.1, SC.2.PE.3.3, ELA.K12.EE.4.1) - Build longer conditional maze solutions as a team. From CodeHopJr to CodeHop Blocks - Navigate the CodeHop editor to create a simple program. From CodeHopJr to CodeHop Blocks: Part 2 - Create a CodeHop program that uses an event and a loop.

Florida Foundations of Computer Science: 2nd Grade 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.
These resources and more are found on the CodeHop Resources Page .	