Florida Unplugged Computer Science 3rd-5th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Florida Unplugged Computer Science 3rd-5th Grade Course** introduces students to foundational digital literacy and programming concepts through unplugged instruction and **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.

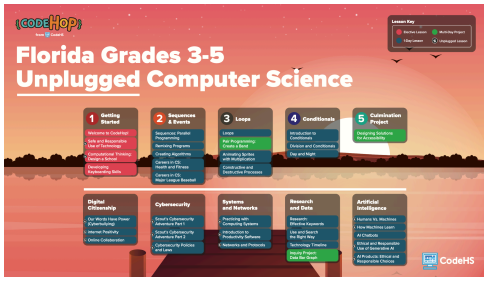
Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts.

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 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. The units in this course are split between programming and digital literacy, providing students with an introduction to a variety of computer science concepts. For classrooms with limited technology, the programming lessons can be completed in small groups or through the use of pair programming, where students work with partners to create a program.

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_Unplugged/overview
https://codehs.com/OK-KCSRoadmap



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

Course Breakdown

Unit 1: Getting Started (2 weeks)

In this introductory unit, students are introduced to the CodeHop Playground and begin developing foundational computational thinking skills.

Objectives / Topics Covered	● Log in and navigate the Playground. ● Apply computational thinking skills. ● Use technology responsibly.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and navigate the Playground to get comfortable using the platform. **Safe and Responsible Use of Technology** ● Learn the importance of balancing screen time and using technology responsibly by creating a personal screen time plan. **Computational Thinking: Design a School** ● Apply computational thinking by breaking down tasks, identifying patterns, and organizing ideas to design a school layout. **Developing Keyboarding Skills (Ongoing typing program)** ● Use proper finger placement to practice typing on a keyboard.

Unit 2: Sequences and Events (5 weeks)

In this unit, students deepen their understanding of sequences and events by designing algorithms and applying coding skills creatively to build interactive projects.

Objectives / Topics Covered	● Create programs using sequences. ● Develop and compare algorithms to solve problems efficiently.
Lessons	**Sequences: Parallel Programming** ● Create a program with multiple sequences running at the same time to control different sprite actions. **Remixing Programs** ● Create or remix digital projects using appropriate content while giving credit to original creators. **Creating Algorithms** ● Write and compare different algorithms to determine which one best fits the goal of the program. **Careers in CS: Health and Fitness** ● Create an animation to demonstrate how technology can be used in health and fitness. **Careers in CS: Major League Baseball** ● Retell important events in a timeline program about how coding can be used in sports.

Unit 3: Loops (5 weeks)

In this unit, students learn to use different types of loops and understand how repetition can simplify code.

Objectives / Topics Covered	● Use different types of loops to simplify algorithms. ● Use repetition to create animation in a program.
Lessons	**Loops** ● Learn that loops repeat one or more instructions and use them in Scratch to simplify and improve programs. **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.

	Animating Sprites with Multiplication ● Use multiplication to animate sprites with loops and wait blocks. **Constructive and Destructive Processes** ● Create an animation that models how volcanoes change surface features through a constructive process.

Unit 4: Conditionals (3 weeks)

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

Objectives / Topics Covered	● Use if/then conditionals to control decision-making in programs. ● Use comparison operators to create more complex decision-making logic.
Lessons	**Introduction to Conditionals** ● Learn how to use if/then blocks to make decisions in a program based on specific conditions. **Division and Conditionals** ● Solve division problems using conditionals. **Day and Night** ● Use operators and conditionals to show how Earth's rotation causes day and night.

Unit 5: Culmination Project (2 weeks)

Students apply their programming knowledge using events, conditionals, and loops to bring their project to life. This final project reinforces key concepts and allows for creativity and problem-solving in a self-directed build.

Objectives / Topics Covered	● Apply key programming concepts including conditionals, loops, and events. ● Demonstrate mastery of coding skills through creative problem-solving.
Lessons	**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 6: Digital Citizenship (3 weeks)

In this unit, students build an understanding of how to be a positive digital citizen. Students will learn how cyberbullying affects others and what actions they can take to maintain a positive online identity.

Objectives / Topics Covered	● Define cyberbullying and explain its consequences. ● Create a code of conduct for responsible internet use. ● Understand how to collaborate with others online.
Lessons	**Our Words Have Power (Cyberbullying)** ● Explain what cyberbullying is and how it affects others. **Internet Positivity** ● Explore how actions can promote positivity online and create a class code of conduct for responsible internet use. **Online Collaboration** ● Understand the value of collaborating online and offline, and how working with others can enhance digital projects through diverse perspectives.

Unit 7: Cybersecurity (3 weeks)

In this unit, students will build cybersecurity skills by exploring common threats, safe habits, and laws that protect people and information online.

Objectives / Topics Covered	• Identify common online threats. • Practice safe online habits. • Protect personal information online.
Lessons	**Scout's Cybersecurity Adventure: Part 1** • Understand basic cybersecurity concepts, identify common online threats, and learn tips for staying safe online. **Scout's Cybersecurity Adventure: Part 2** • Practice safe online habits and learn about tools and technologies that help protect personal information. **Cybersecurity Policies and Laws** • Explore how cybersecurity policies apply in the classroom and research a state-specific cybersecurity law.

Unit 8: Systems and Networks (3 weeks)

In this unit, students develop an understanding of computing systems and networks, exploring troubleshooting strategies, types of software, and learning how networks connect devices.

Objectives / Topics Covered	• Troubleshoot basic computer problems. • Compare software applications to complete different tasks. • Investigate how networks transfer data.
Lessons	**Practicing with Computing Systems** • Interact with and troubleshoot basic computing systems. **Introduction to Productivity Software** • Compare and select software applications to complete different tasks. **Networks and Protocols** • Learn how network protocols work, how data is transferred, and compare the functions of WiFi, wired, and cellular networks.

Unit 9: Research and Data (5 weeks)

In this unit, students will develop research skills such as using effective keywords to search for information, giving credit for sources found online, collecting and interpreting data, and sharing conclusions with others.

Objectives / Topics Covered	• Search for information using keywords. • Cite research sources. • Gather and interpret data.
Lessons	**Research: Effective Keywords** • Evaluate the effectiveness of different keywords when using a search engine to determine if a source is credible and relevant. **Use and Search the Right Way** • Research information online to answer questions and give credit to sources properly. **Technology Timeline** • Create an interactive timeline to illustrate key developments in music player technology. **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.

Unit 10: Artificial Intelligence (5 weeks)

In this unit, students will explore current AI technologies and explore how these technologies learn and reason. Students will also discuss current ethical issues around using AI responsibly.

Objectives	• Compare human and computer performance on tasks.

/ Topics Covered	• Explain different machine learning approaches. • Analyze the benefits and challenges of AI products.
Lessons	**Humans Vs. Machines** • Compare and contrast human and computer performance on similar tasks. **How Machines Learn** • Explain different machine learning approaches. **AI Chatbots** • Describe how AI representations support reasoning by training a model AI chatbot program. **Ethical and Responsible Use of Generative AI** • Describe the pros and cons of generative AI and complete a class Code of Conduct to follow when using AI. **AI Products: Ethical and Responsible Choices** • Describe how AI products work and analyze the benefits and challenges of various AI products from multiple perspectives.

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 **CodeHop Resources Page**.	