Indiana 4th Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Indiana 4th Grade Computer Science 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 lessons**, each approximately 45 minutes long. This provides a full school year of material if teaching one lesson per week. 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/22929/overview?lang=en

A clickable PDF can be found at https://codehs.com/IN-CSRoadmaps.

Course Breakdown

Unit 1: Optional Review (3 weeks)

In this unit, students review core skills by logging into CodeHop, exploring coordinate-based animation in Scratch, and customizing their projects with creative drawing tools.

Objectives / Topics Covered	● Log in and navigate the CodeHop Playground. ● Use the coordinate plane to position and animate sprites in Scratch. ● Create custom sprites and backdrops using Scratch's drawing tools.
Lessons	**Welcome to CodeHop!** ● Log in and explore how to use the CodeHop Playground as an introduction or warm-up. **The Coordinate Plane** ● Create an open-ended animation using the coordinate plane to position and move sprites. **Scratch Drawing Tools** ● Use Scratch's drawing tools to design custom sprites and backdrops for creative projects.

Unit 2: Getting Started (2 weeks)

In this unit, students explore the components of computing systems and reflect on how to contribute to a positive and responsible online environment.

Objectives / Topics Covered	● Identify hardware and software components of a computing system. ● Recognize and respond to simple technology problems. ● Promote responsible, positive behavior online through a personal code of conduct.
Lessons	**Exploring Computing Systems** ● Identify parts of a computing system and troubleshoot basic hardware and software issues. **Internet Positivity** ● Discuss how online actions affect others and create a code of conduct for positive internet behavior.

Unit 3: Sequences and Events (6 weeks)

In this unit, students apply sequencing and event-based programming to create interactive projects, collaborate through pair programming, and use broadcast messages to coordinate sprite interactions.

Objectives / Topics Covered	● Apply computational thinking to design creative solutions. ● Use multiple types of event blocks and program algorithms in Scratch. ● Collaborate through pair programming and coordinate sprites using broadcasts.
Lessons	**Computational Thinking: Design a School** ● Use computational thinking strategies to design and model a school layout or system. **Events: Dot in Space** ● Create a program that uses different types of event blocks to animate a sprite. **Creating Algorithms** ● Program and compare multiple algorithms to determine which works best for a given task. **Pair Programming: Create a Band (2-part lesson)** ● Collaborate using pair programming to design a Scratch band project that responds to

	keyboard inputs. **Broadcast Messages: Tell a Joke** ● Use broadcast messages to program a knock-knock joke between two sprites.

Unit 4: Loops (2 weeks)

In this unit, students explore how loops repeat instructions in Scratch and practice debugging programs that use events and loops.

Objectives / Topics Covered	● Use loops to repeat one or more instructions in a Scratch program. ● Decompose and debug programs that combine loops and events.
Lessons	**Loops** ● Explain how loops repeat instructions and use them to simplify code in a Scratch project. **Debugging: Events and Loops** ● Decompose and fix a program that uses both events and loops to ensure it runs as intended.

Unit 5: Conditionals (5 weeks)

In this unit, students expand their understanding of conditionals by creating interactive programs and games that use if/then logic and multiple conditions to control behavior.

Objectives / Topics Covered	● Use if/then conditionals to guide sprite behavior in Scratch. ● Apply conditional logic in creative, interactive projects. ● Combine conditionals with other programming concepts to build functional games.
Lessons	**Conditionals: Underwater Exploration** ● Create a Scratch program that uses conditionals to control actions in an underwater setting. **Scout's Quest: Conditionals** ● Use if/then conditionals to complete the final challenge in the Scout's Quest skill review series. **Conditionals: Flying Bird (2-part lesson)** ● Program a Flying Bird game using multiple conditionals to control gameplay. **Game Effects** ● Modify a game to add engaging effects and make updates based on peer feedback.

Unit 6: Variables and Lists (4 weeks)

In this unit, students use variables and lists to track and manage data in Scratch, applying these concepts in games and interactive projects.

Objectives / Topics Covered	● Create and update variables to track information like scores. ● Use lists to organize and reference data in a program. ● Apply variables and lists in game-based projects for dynamic interactions.
Lessons	**Scout's Quest: Variables** ● Create and use variables to track points in a program as part of the Scout's Quest review series. **Lists: Spelling Bee** ● Use lists in Scratch to build an interactive spelling bee game. **Pong Game (2-part lesson)** ● Create a pong-style game that uses variables to track player scores.

Unit 7: Clones and Functions (5 weeks)

In this unit, students explore how to use clones and functions in Scratch, incorporating variables, Boolean logic, and inputs to build interactive animations, games, and custom blocks.

Objectives / Topics Covered	● Create and manage clones to duplicate sprites in animations and games. ● Use variables and clones to build a functional snake game. ● Design functions with Boolean and number inputs to create flexible, reusable code blocks.
Lessons	**Introduction to Clones** ● Create an animation using clones and explore the limits of how clones behave in Scratch. **Snake Game (2-part lesson)** ● Build a snake game using clones and variables to control gameplay and track data. **Scout's Quest: Functions with Boolean Inputs** ● Create a function with a Boolean input to trigger different actions based on password correctness. **Scout's Quest: Functions with Number Inputs** ● Use a function with number inputs to generate drawings programmatically.

Unit 8: Culmination Projects (6 weeks)

In this unit, students apply their understanding of Scratch programming concepts to design original projects that combine interactivity, creativity, and cross-curricular content.

Objectives / Topics Covered	● Build interactive games and tools using conditionals, variables, booleans, and events. ● Apply creative expression and logic to custom game and music player projects. ● Integrate cultural understanding into a branching narrative experience.
Lessons	**Click-a-Mole (2-part lesson)** ● Create a Whack-a-Mole style game using conditionals, variables, booleans, and event blocks. **Code Tunes (2-part lesson)** ● Design a custom Scratch music player using variables, operators, and conditionals. **Choose Your Own Path: Elements of Culture (2-part lesson)** ● Create a choose-your-own-path game that highlights cultural elements through interactive storytelling.

Unit 9: Digital Literacy (6 weeks)

In this unit, students build digital literacy by exploring digital identity, understanding how data moves through networks, and learning foundational cybersecurity concepts to stay safe online.

Objectives / Topics Covered	● Explain how information travels through the Internet using packets. ● Practice safe online habits and understand cybersecurity tools. ● Compare human and computer performance and limitations. ● Collect and display data using visual representations.
Lessons	**Impacts of Computing: Exploration** ● *Coming soon!* **Inquiry Project: Data Bar Graph (2-part lesson)** ● Follow the inquiry process to collect data and modify a Scratch program to display results in a bar graph. **Humans Vs. Machines** ● Compare human and computer strengths, discuss technology's limitations, and examine how computers perceive the world.

	Scout's Cybersecurity Adventure: Part 2 ● Learn and demonstrate online safety by exploring cybersecurity tools and habits that protect personal information. **Networks, Packets, and the Internet** ● Explain how data is sent over the Internet using packets, and create a secure method for classroom communication.

4th - 5th 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.
These resources and more are found on the **Elementary Resources Page**.	