Indiana Kindergarten Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Indiana Kindergarten Computer Science 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 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 **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/22913/overview?lang=en.

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

Course Breakdown

Unit 1: Optional Unplugged Exploration (2 weeks)

In this unit, students begin learning foundational programming concepts by creating and following sequences of instructions through movement-based activities and collaborative games.

Objectives / Topics Covered	● Create ordered sequences of instructions. ● Apply sequencing to solve simple problems collaboratively.
Lessons	**Sequences (Unplugged)** ● Create a sequence of step-by-step instructions to choreograph a dance. **Coding Card Game: Sequences** ● Work together to build a sequence of instructions to guide Scout through a maze.

Unit 2: Getting Started (5 weeks)

In this unit, students are introduced to basic computer operations and responsible use, including logging in, using input devices, identifying computer parts, and practicing safe technology habits.

Objectives / Topics Covered	● Log in and navigate the CodeHop Playground. ● Identify basic computer components and functions. ● Practice using the mouse and keyboard. ● Understand and follow safe, responsible technology use.
Lessons	**Welcome to CodeHop!** ● Log in and explore how to use the CodeHop Playground in a short introductory session. **Computer Basics: Introduction** ● Learn what a computer is, how we use it, and how to identify input, output, hardware, and software. **Mouse Practice** ● Demonstrate mouse skills through clicking and dragging in interactive games. **Keyboard Introduction** ● Practice using letters, numbers, and basic keyboard functions. **Introduction to Responsible Technology Use** ● Identify safe and responsible ways to use technology, including understanding an Acceptable Use Policy (AUP).

Unit 3: ScratchJr Exploration (6 weeks)

In this unit, students follow a sequential, story-based adventure to learn ScratchJr programming skills, including adding characters and scenes, using motion blocks, and building animated stories with sequences and events.

Objectives / Topics Covered	● Navigate the ScratchJr interface and add or modify characters and scenes. ● Use motion blocks and sequences to animate actions. ● Combine elements across multiple pages using events for storytelling.
Lessons	**Scout Adventures 1: Introducing Scout** ● Explore the ScratchJr interface and add characters to the stage. **Scout Adventures 2: Scout Starts Exploring** ● Add backgrounds and a new page to begin a multi-scene story. **Scout Adventures 3: Scout Meets a Friend** ● Delete and modify characters to develop the narrative. **Scout Adventures 4: Scout Explores the Forest**

	● Use motion blocks to move characters around the stage. **Scout Adventures 5: Scout and Bluebird Help** ● Build a sequence of motion blocks to guide characters in collecting objects. **Scout Adventures 6: Scout Celebrates with Friends** ● Create a celebration scene using characters, pages, backgrounds, sequences, and events.

Unit 4: Sequences and Events (6 weeks)

In this unit, students build on sequencing skills by exploring events, visibility, and size-changing blocks in ScratchJr, while applying computational thinking to routines and creative projects.

Objectives / Topics Covered	● Apply sequencing and decomposition to everyday routines. ● Use ScratchJr tools like painting, events, show/hide, and size-changing blocks. ● Create interactive scenes by combining visual effects and event triggers.
Lessons	**Computational Thinking: Morning Routines** ● Use computational thinking to identify patterns, sequence steps, and simplify morning routines. **Drawing Tools: Fairy Tale Painting** ● Use ScratchJr painting tools to design a custom fairy-tale scene. **Introduction to Events** ● Create a program that responds to different types of events. **Introduction to Show and Hide Blocks** ● Use “show” and “hide” blocks in a sequence to control character visibility. **Introduction to Grow and Shrink Blocks** ● Build a program that changes character size using “grow” and “shrink” blocks. **Growing Garden** ● Model a garden growing by using grow, shrink, and hide blocks in sequence.

Unit 5: Pages (3 weeks)

In this unit, students learn to build multi-page programs in ScratchJr and use navigation blocks to move between different scenes.

Objectives / Topics Covered	● Create multi-page stories in ScratchJr. ● Use the “go to page” block to connect and navigate between pages.
Lessons	**Introduction to Pages** ● Create a program that includes multiple pages to expand a story or project. **Pages: Dragon Story** ● Build a multi-page ScratchJr program to tell a simple dragon-themed story. **Using the Go To Page Block** ● Use the “go to page” block to move between pages within a program.

Unit 6: Block Exploration (5 weeks)

In this unit, students explore ScratchJr blocks that control speed, sound, and motion, while creating personalized programs and practicing debugging skills.

Objectives / Topics Covered	● Use speed and sound blocks to animate characters. ● Personalize programs with creative content. ● Identify and fix errors in event- and motion-based sequences.
Lessons	**Introduction to Speed Blocks** ● Create a program that uses different speed blocks to animate character movements.

	All About Me! (2-part lesson) ● Design a personalized program that shares favorite things using characters and scenes. **Debugging: Events and Motion** ● Find and fix bugs in sequences involving events and motion blocks. **Transportation Speeds and Sounds** ● Use "say," "sound," and speed blocks to program various modes of transportation.

Unit 7: Loops (3 weeks)

In this unit, students learn to use loops in ScratchJr to repeat actions efficiently, including sequences that run continuously with forever loops.

Objectives / Topics Covered	● Use loops to repeat actions in a program. ● Create repeating sequences, including forever loops for continuous animation.
Lessons	**Loops** ● Create a program using loops and explain how they help repeat code efficiently. **Loops: Frog and Rabbit** ● Use loops to repeat actions in a simple character-based program. **Forever Loops: Fireworks** ● Create a program with forever loops to animate a continuous fireworks display.

Unit 8: Culmination Projects (4 weeks)

In this unit, students apply their coding knowledge to create original ScratchJr projects that combine interaction, animation, and storytelling elements.

Objectives / Topics Covered	● Program interactive characters using events, sequences, and loops. ● Design creative projects that showcase learned ScratchJr skills.
Lessons	**Chicken Crossing Game (2-part lesson)** ● Program a game with a player-controlled character and adjustable difficulty. **Wildlife Scene Project (2-part lesson)** ● Create a wildlife scene using events, sequences, and loops to animate the environment.

Unit 8: Digital Literacy (5 weeks)

In this unit, students build digital literacy by exploring how to find, collect, and store information responsibly while learning about data, privacy, and visual communication.

Objectives / Topics Covered	● Conduct basic research and present findings visually. ● Collect, organize, and analyze simple data. ● Understand how computers store data and protect personal information.
Lessons	**Introduction to Research (2-part lesson)** ● Find information using research sources and create a visual ScratchJr program to share it. **What Can Data Tell Us?** ● Collect and organize data about school transportation, then analyze it to draw conclusions. **Introduction to Data Storage and Files** ● Learn that computers store data as files and model how data is collected and stored. **Keeping Information Safe** ● Identify and understand the difference between private and personal information to

	stay safe online.

K-1 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.
These resources and more are found on the **Elementary Resources Page**.	