Indiana 5th Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Indiana 5th 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/22930/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, reviewing computer science vocabulary, and exploring coordinate-based animation in Scratch.

Objectives / Topics Covered	● Log in and navigate Scratch and the CodeHop Playground. ● Use the coordinate plane to position and animate sprites in Scratch.
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. **Introduction to Computer Science and Scratch** ● Define important computer science vocabulary and create a simple program.

Unit 2: Getting Started (2 weeks)

In this unit, students review computing system components and learn how to recognize and respond to cyberbullying and other harmful online behaviors.

Objectives / Topics Covered	● Identify basic hardware and software components of a computing system. ● Troubleshoot simple technology problems. ● Recognize cyberbullying and describe appropriate ways to respond.
Lessons	**Practicing with Computing Systems** ● Identify parts of a computing system and address basic hardware and software issues. **Standing Up to Cyberbullying** ● Recognize harmful online behaviors, including cyberbullying, and explore strategies for response and prevention.

Unit 3: Sequences and Events (3 weeks)

In this unit, students apply sequencing and event-based programming while using computational thinking to solve problems and debug Scratch programs.

Objectives / Topics Covered	● Apply computational thinking to design solutions. ● Use events to control program behavior. ● Debug programs by identifying and fixing errors.
Lessons	**Computational Thinking: Design a Neighborhood** ● Use computational thinking strategies to plan and create a neighborhood layout. **Events: Traveling with Scout** ● Program character actions using events to control timing and behavior. **Debugging: Make a Pizza** ● Debug a program by breaking it down into steps and correcting errors in a pizza-making sequence.

Unit 4: Loops (3 weeks)

In this unit, students explore how loops can be used in Scratch to create artistic patterns and multi-scene animations.

Objectives / Topics Covered	● Use loops to create repeating patterns and movements. ● Apply repeat loops to build multi-scene animations.
Lessons	**Creating Turtle Graphics** ● Use the pen tool and loops in Scratch to draw turtle-style geometric patterns. **Animation Loops Project (2-part lesson)** ● Create a multi-scene animation using repeat loop blocks to drive character actions.

Unit 5: Conditionals and Operators (4 weeks)

In this unit, students explore conditionals and operators by designing interactive Scratch programs that involve decision-making, planning, and randomized outcomes.

Objectives / Topics Covered	● Use conditionals to guide program decisions. ● Decompose and plan the steps for an interactive project. ● Apply operators and variables to create randomized outcomes.
Lessons	**Conditionals: Mazes** ● Create a Scratch program that uses conditionals to navigate through a maze. **Plan a Quest (2-part lesson)** ● Plan and break down the steps required to create a quest-based interactive program. **Operators: Coin Flip** ● Use operators and variables to program a coin flip with randomized results.

Unit 6: Variables and Lists (9 weeks)

In this unit, students build more advanced game and simulation projects by using variables, lists, operators, and object-based thinking to create dynamic, interactive experiences in Scratch.

Objectives / Topics Covered	● Use variables and operators to control behaviors. ● Apply lists and conditionals to simulate interactive choices. ● Explore comparison operators and randomness to drive game mechanics. ● Understand basic class and object concepts in programming.
Lessons	**Variables in Dance** ● Use variables to control the pitch and speed of dance moves in a Scratch animation. **Lists: Shopping with Scout (2-part lesson)** ● Create a shopping simulator using variables, lists, and operators to track selections. **Game Mechanics with Comparison Operators** ● Use comparison operators with variables to create winning or losing game conditions. **Pinball Game Project (3-part lesson)** ● Design and build a functional pinball game using core game design principles. **Classes and Objects in Games (2-part lesson)** ● Learn about classes and objects by creating a game where randomizers modify object characteristics.

Unit 7: Clones and Functions (4 weeks)

In this unit, students use clones to efficiently create game elements and explore how functions with inputs can simplify and personalize their Scratch programs.

Objectives / Topics Covered	● Use clones to build dynamic, reusable elements in games. ● Create and use functions with inputs to organize code. ● Understand the role of clones and functions in efficient program design.
Lessons	**Clones: Throwing Acorns Game** ● Create a game that uses clones to throw multiple acorns at targets. **Clones in Games (2-part lesson)** ● Program an endless runner game using clones and explain how clones support game efficiency. **Functions: About Me** ● Create and use a function with input to personalize a Scratch program.

Unit 8: Culmination Projects (6 weeks)

In this unit, students apply their programming knowledge and design thinking skills to create original projects that showcase creativity, problem-solving, and technical mastery.

Objectives / Topics Covered	● Design and build a game using loops, conditionals, variables, and other core skills. ● Apply the design thinking process to create a functional app that addresses a real-world need. ● Demonstrate mastery of programming concepts through independent project work.
Lessons	**Game Design Project (3-part lesson)** ● Design and create a game that incorporates loops, conditionals, variables, and interactive gameplay elements. **Design an App (3-part lesson)** ● Follow the design thinking process to build an app that solves a user-centered problem.

Unit 9: Digital Literacy (5 weeks)

In this unit, students expand their digital literacy by exploring data visualization, network protocols, machine learning, attribution, and cryptography through interactive and analytical activities.

Objectives / Topics Covered	● Visualize data to support claims and communicate findings. ● Understand how network protocols and connection types enable communication. ● Explore machine learning and classification through decision trees. ● Practice ethical digital behavior through attribution and encryption.
Lessons	**Using Digital Tools to Create Line Graphs** ● Convert table data into a line graph to support and explain a specific claim. **Networks and Protocols** ● Explain how data moves across networks and compare WiFi, wired, and cellular connection types. **How Machines Learn** ● Explore different types of machine learning and create a classification system using a decision tree. **Giving Credit Through Attributions** ● Learn how to give proper credit when using or remixing digital content. **Scout's Cryptography Escape Room** ● Solve ciphers and learn basic cryptographic techniques in an interactive escape room experience.

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**.	