Indiana 2nd Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Indiana 2nd Grade 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/22925/overview?lang=en.

Course Breakdown

Unit 1: Optional Review (6 weeks)

In this unit, students review key programming concepts such as events and loops while practicing ScratchJr skills like scene creation, character interaction, and message-based coordination.

Objectives / Topics Covered	● Navigate the ScratchJr interface and build animated scenes. ● Use event blocks and loops, including repeat and forever loops. ● Program character interactions using message events.
Lessons	**Welcome to CodeHop!** ● Log in and explore how to use the CodeHop Playground in a short introductory session. **Introduction to ScratchJr** ● Navigate the ScratchJr interface and create a scene with characters. **Events** ● Explain the concept of events in programming and use multiple event blocks in a project. **Introduction to Repeat Loops** ● Use repeat loops to run a section of code several times. **Forever Loop Dance Party** ● Create a sequence using a “repeat forever” loop to make characters continuously repeat actions. **Introduction to Message Events** ● Program a relay race in ScratchJr where characters interact by sending and receiving messages.

Unit 2: Getting Started (3 weeks)

In this unit, students review computer fundamentals and learn how to be responsible digital citizens by understanding how technology works and how to use it safely.

Objectives / Topics Covered	● Identify computer components and how they work together. ● Understand digital citizenship, including safety, cyberbullying, and digital footprints. ● Practice responsible use of technology in school settings.
Lessons	**Computer Basics: Connections** ● Learn what a computer is, how it works, and how input, output, hardware, and software function together. **Responsible Digital Citizens** ● Explore what it means to be a responsible digital citizen, including digital footprints, cyberbullying, and how to report concerns. **Practicing Responsible Technology Use** ● Demonstrate safe and respectful technology use through everyday examples.

Unit 3: Sequences and Events (7 weeks)

In this unit, students apply sequencing and events to real-world and creative contexts, while practicing debugging, spatial reasoning, collaboration, and exploring careers in computer science.

Objectives / Topics Covered	● Apply computational thinking to routines and sequences. ● Use events to build interactive programs and games. ● Debug code, use the grid for movement, and collaborate with peers.

	● Explore how coding supports careers like fashion design.
Lessons	**Computational Thinking: School Day Routines** ● Use computational thinking to identify patterns, sequence steps, and simplify parts of the school day. **Debugging: Events and Sequences** ● Find and fix errors in sequences and event-based programs. **Tap-a-Mole Game (2-part lesson)** ● Create an interactive game using tap events. **Introduction to the Grid** ● Use the ScratchJr grid to move characters to specific locations on the stage. **Careers in CS: Coding for Fashion-Retail** ● Learn how coding supports fashion design and create a program to design and animate a fashion character. **Two-Step Dance & Feedback** ● Create a dance-themed program, revise it based on peer feedback, and give attribution for help received.

Unit 4: Message Events (3 weeks)

In this unit, students explore how message events can be used to control program flow, model cycles, and connect actions across multiple pages in ScratchJr.

Objectives / Topics Covered	● Use message events to trigger actions in sequence. ● Model repeating or connected processes using message-based logic. ● Link pages and events to create multi-step programs.
Lessons	**Message Events: Scout Plays in the Forest** ● Use message events to control the order of actions in a ScratchJr program. **Programming a Cycle** ● Model a repeating cycle by sending and receiving messages between characters. **Pages: Scout's Travels** ● Help Scout move between pages using message events to connect scenes.

Unit 5: Loops (5 weeks)

In this unit, students use loops to identify patterns, control timing, and animate stories, while debugging more complex programs involving message events and repeated actions.

Objectives / Topics Covered	● Create and debug programs using loops and message events. ● Use loops with wait and turn blocks to control timing. ● Animate original stories using sequences and repeated actions.
Lessons	**Loops: Follow the Path** ● Identify patterns and create a program that uses loops to repeat movement along a path. **Debugging: Message Events and Loops** ● Find and fix problems in code that uses both message events and loops. **Making a Timer** ● Use loops, wait blocks, and turn blocks to build and compare two animated timers. **Original Story Animations – Fiction Story (2-part lesson)** ● Develop an original fiction story and animate it using ScratchJr programming tools.

Unit 6: Culmination Projects (12 weeks)

In this unit, students synthesize their learning by designing interactive games and visual projects that combine sequences, events, loops, messages, and data collection.

Objectives / Topics Covered	- Design interactive games using core ScratchJr programming concepts. - Apply sequences, loops, events, and messages across multi-page projects. - Collect and visualize data in a creative program.
Lessons	**Moving Targets Game (3-part lesson)** - Create a game with moving targets using sequences, events, and multiple pages. **Maze Game Project (3-part lesson)** - Build and refine a maze game using events, messages, loops, and sequences, incorporating peer feedback. **Design an Adventure Game (3-part lesson)** - Create a multi-page, story-driven adventure game using a combination of learned coding skills. **Advanced Data and Programming (3-part lesson)** - Develop a question, gather data through a survey, and present the results using a ScratchJr program.

Unit 7: Digital Literacy (6 weeks)

In this unit, students develop digital literacy by exploring data storage, safe login practices, software use, data interpretation, and research skills.

Objectives / Topics Covered	- Understand secure login practices using usernames and passwords. - Recognize how computers store and manage data. - Interpret data patterns and present predictions using visual programs. - Conduct research using reliable sources and communicate findings. - Compare software tools for different computing tasks.
Lessons	**Password Protectors** - Learn the importance of usernames and passwords and how to keep login information secure. **Managing Data Storage and Files** - Understand how computers store information as files and model data collection and storage. **Data Patterns and Predictions** - Identify patterns in data visualizations and create a program to communicate trends and predictions. **Choice Research (2-part lesson)** - Collect and evaluate sources to answer a research question and share findings through a visual program. **Types of Software** - Compare different types of software and choose appropriate tools for completing specific tasks.

Unit 8: Optional: Preparing for Next Year (5 weeks)

In this unit, students prepare for more advanced programming by exploring conditionals, variables, and transitioning from ScratchJr to Scratch with foundational event and loop concepts.

Objectives / Topics Covered	- Use conditionals and variables in unplugged and digital activities. - Explore the Scratch interface and build simple programs. - Apply event and loop knowledge when transitioning from ScratchJr to Scratch.
Lessons	**Coding Card Game: Conditionals** - Work collaboratively to guide Scout through a maze using sequences with conditionals. **Coding Card Game: Conditionals 2** - Extend conditional logic in a second maze activity with more complex instructions. **Variables: Keeping Score**

	● Create a program that simulates scoring by using a variable. **From ScratchJr to Scratch** ● Navigate the basic Scratch interface and create a simple introductory program. **ScratchJr to Scratch: Events and Loops** ● Build a Scratch program that includes both an event and a loop to animate actions.

2nd 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.
These resources and more are found on the **Elementary Resources Page**.	