Indiana 1st Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Indiana Computer Science 1st Grade Course introduces students to foundational programming concepts through 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 teacher-led and includes ready-to-use lessons following a consistent structure: Introduction, Guided Practice, Independent Practice, Extension, and Reflection. Instruction follows an "I do, we do, you do" model and incorporates spiral review to reinforce concepts and build confidence over time.

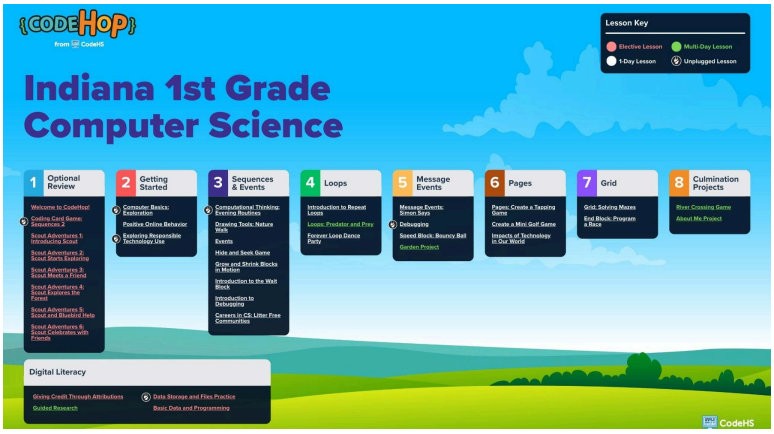
The course includes 36 lessons, each approximately 45 minutes long, providing a full year of instruction when taught once per week. While the course allows for instructional flexibility, some lessons are required to fully meet state computer science standards and are clearly identified within the syllabus. All Digital Literacy lessons are required to ensure full standards alignment, as they address essential non-programming computer science concepts. All lessons are labeled with the specific standards they address to support planning and compliance.

Standards Alignment Note: Standards addressed in each lesson are listed in parentheses after the lesson title. Standards may be reinforced across multiple lessons, and full coverage of state computer science standards is achieved through the course as a whole.

Programming Environment: Students will write and run programs that are 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/22921/overview?lang=en



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

Course Breakdown

Optional Review (7 lessons)

This foundational set of lessons introduces students to the programming interface and collaborative storytelling through sequential, narrative-driven projects.

Objectives / Topics Covered	● Programming interface basics ● Collaborative storytelling ● Sequential programming projects
Lessons	**Welcome to CodeHop! (K-2.DI.2)** ● Log in and explore how to use the CodeHop Playground. **Coding Card Game: Sequences 2 (K-2.PA.1, K-2.PA.5)** ● Work together to create a sequence of instructions to move Scout through a maze. **Scout Adventures 1: Introducing Scout (K-2.CD.3)** ● Explore the programming interface and add characters. **Scout Adventures 2: Scout Starts Exploring (K-2.CD.3)** ● Add backgrounds and a page to a program. **Scout Adventures 3: Scout Meets a Friend (K-2.CD.3)** ● Delete and modify characters in a program. **Scout Adventures 4: Scout Explores the Forest (K-2.CD.3)** ● Explore and use motion blocks to move characters around the stage. **Scout Adventures 5: Scout and Bluebird Help (K-2.CD.3)** ● Build a sequence of motion blocks to move characters to collect objects. **Scout Adventures 6: Scout Celebrates with Friends (K-2.CD.3)** ● Create a celebration scene by adding characters, pages, backgrounds, and sequences of motion blocks with events.

Unit 1: Getting Started (4 lessons)

Students explore computer basics, learn to identify hardware and software, and practice safe, responsible online behavior.

Objectives / Topics Covered	● Computer hardware and software ● Input and output concepts ● Online safety and responsible behavior
Lessons	**Computer Basics: Exploration (K-2.CD.1, K-2.CD.2)** ● Learn what a computer is, how to use it, and what to do when it doesn't work. **Positive Online Behavior (K-2.NI.1, K-2.IC.2, K-2.IC.3)** ● Identify appropriate and inappropriate online behavior. **Exploring Responsible Technology Use (K-2.NI.1, K-2.IC.2, K-2.IC.3)** ● Explain ways to use technology safely and responsibly. **Computational Thinking: Evening Routines (K-2.PA.5)** ● Use computational thinking to identify patterns, break down tasks, sequence steps, and simplify processes.

Unit 2: Sequences & Events (7 lessons)

This unit teaches students to use drawing tools, event blocks, and debugging techniques to build interactive games and animations.

Objectives / Topics Covered	● Drawing tool applications ● Event-based programming ● Interactive game creation ● Basic debugging techniques
Lessons	**Drawing Tools: Nature Walk (K-2.PA.1)** ● Use drawing tools to create a nature-walk scene. **Events (K-2.PA.1, K-2.PA.3, K-2.PA.4, K-2.PA.5)** ● Explain what an event is and use multiple event blocks. **Hide and Seek Game (K-2.PA.4)** ● Use the hide block to program an interactive game of hide-and-seek. **Grow and Shrink Blocks in Motion (K-2.PA.1, K-2.PA.3)** ● Create a program using motion blocks and grow and shrink blocks to change character size. **Introduction to the Wait Block (K-2.PA.3)** ● Use "wait" blocks to cause characters to pause. **Introduction to Debugging (K-2.PA.1, K-2.PA.2, K-2.PA.4)** ● Find and correct bugs in sequences. **Careers in CS: Litter Free Communities (K-2.PA.1, K-2.PA.3, K-2.IC.1)** ● Explain how computer science solves problems and create a program to collect and sort litter.

Unit 3: Loops (4 lessons)
Students learn to optimize their code by using repeat loops and forever loops to create efficient sequences and interactive programs.

Objectives / Topics Covered	● Implementing repeat loops ● Creating efficient sequences ● "Repeat forever" programming
Lessons	**Introduction to Repeat Loops (K-2.PA.1, K-2.PA.3, K-2.PA.4)** ● Use repeat loops to run a section of code multiple times. **Loops: Predator and Prey (2 classes | K-2.PA.3)** ● Use events, sequences, and loops to program two animal characters to interact. **Forever Loop Dance Party (K-2.PA.1, K-2.PA.3, K-2.PA.4, K-2.PA.5)** ● Create a sequence using a "repeat forever" loop.

Unit 4: Message Events (4 lessons)
This unit focuses on using message blocks and debugging to coordinate interactions between multiple characters in complex projects.

Objectives / Topics Covered	● Coordinating character interactions ● Using message blocks ● Debugging complex logic
Lessons	**Message Events: Simon Says (K-2.PA.1, K-2.PA.3)** ● Use message events to make one character communicate to many. **Debugging (K-2.PA.1, K-2.PA.2, K-2.PA.4)** ● Describe what bugs are and find and correct bugs in sequences. **Speed Block: Bouncy Ball (K-2.PA.1, K-2.PA.3, K-2.PA.4)** ● Use speed blocks and messages to program movement at different speeds. **Garden Project (K-2.PA.3, K-2.PA.4)**

	● Use message events, grow, shrink, hide, and show blocks to animate seeds growing.

Unit 5: Pages (3 lessons)
Students expand their programming skills by creating multi-page games and exploring the broader impacts of technology on our world.

Objectives / Topics Covered	● Creating multi-page projects ● Navigation between scenes ● Technology's societal impact
Lessons	**Pages: Create a Tapping Game (K-2.PA.3)** ● Create a game that moves from one page to the next. **Create a Mini Golf Game (K-2.PA.3)** ● Use messages and loops to create a mini golf game. **Impacts of Technology in Our World (K-2.IC.1)** ● Create a program to demonstrate how technology impacts our world.

Unit 6: Grid (2 lessons)
This unit introduces grid-based movement, allowing students to design custom mazes and program characters to navigate them effectively.

Objectives / Topics Covered	● Grid-based movement logic ● Custom maze design ● Animation control
Lessons	**Grid: Solving Mazes (K-2.PA.1, K-2.PA.3, K-2.PA.4)** ● Design a maze and use the grid to program a character to move through it. **End Block: Program a Race (K-2.PA.1, K-2.PA.3, K-2.PA.4)** ● Program a race and use the "end" block in an animation.

Optional Culmination Projects (2 lessons)
Students apply their cumulative knowledge of sequencing, loops, and messages to build advanced, feature-rich games and projects.

Objectives / Topics Covered	● Advanced project design ● Applying sequencing and loops ● Feature-rich game development
Lessons	**River Crossing Game (K-2.PA.3, K-2.PA.4)** ● Program obstacles in a game and change difficulty using speed blocks. **About Me Project (K-2.PA.1, K-2.PA.3)** ● Plan a program to describe characteristics and interests.

Digital Literacy (4 lessons)
This unit covers essential non-programming concepts, including digital research, data storage, and the importance of giving proper credit to others' work.

Objectives / Topics Covered	● Digital research methods ● Data collection and storage ● Giving credit and attribution ● Digital citizenship

Lessons	**Giving Credit Through Attributions (K-2.PA.6)** ● Explain why giving credit is important and reuse parts of a story with attribution. **Guided Research (2 classes | K-2.DI.1, K-2.DI.2, K-2.PA.3, K-2.PA.6)** ● Find information using research sources and create a program to communicate research visually. **Data Storage and Files Practice (K-2.DI.2, K-2.DI.3)** ● Recognize that computers store data as files and model how data is collected. **Basic Data and Programming (K-2.DI.1, K-2.DI.2, K-2.DI.4)** ● Collect data and create a program to present data visually.

1st 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**.	