Idaho Computer Science 5th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Idaho Computer Science 5th Grade 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**, with each lesson approximately 45 minutes long. This provides a full school year of material if teaching one lesson per week. Optional 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 the CodeHS platform. 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/26582/overview

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

Course Breakdown

Optional Review

In this foundational unit, students get reacquainted with CodeHop and begin to build familiarity with Scratch's interface, coordinate systems, and creative tools.

Objectives / Topics Covered	● Log in and navigate the CodeHop Playground. ● Understand computer science terms and programming basics. ● Explore Scratch's coordinate plane and drawing tools.
Lessons	**Welcome to CodeHop!** ● Practice logging in and exploring the Playground before starting a full lesson. **Introduction to Computer Science and Scratch** ● Review basic computer science vocabulary and create a simple Scratch program. **The Coordinate Plane** ● Use the coordinate plane in Scratch to design an open-ended animation. **Scratch Drawing Tools** ● Create customized sprites and backdrops using the drawing tools.

Unit 1: Sequences & Events (7 lessons)

Students practice algorithmic thinking and event-based programming while exploring careers, refining their code, and using creative tools like the pen extension.

Objectives / Topics Covered	● Design creative sequences using events. ● Refine multiple algorithms for efficiency and clarity. ● Apply drawing tools and pen graphics in programming. ● Explore real-world CS applications in sports.
Lessons	**Computational Thinking: Design a Neighborhood** ● Apply computational thinking to plan and design a neighborhood layout. **Drawing Tools: Sea Creatures** ● Use Scratch image editing tools to create and program deep-sea creatures. **Events: Traveling with Scout** ● Use event blocks to trigger character actions in a travel-themed program. **Compare and Refine Algorithms** ● Evaluate and improve multiple algorithms for a task. **Debugging: Make a Pizza** ● Break down and fix a pizza-making program to ensure it runs correctly. **Careers in CS: Major League Baseball** ● Learn how coding is used in sports and build a timeline program to retell key events from an article. **Creating Turtle Graphics** ● Use loops and the pen tool in Scratch to draw repeated, artistic patterns.

Unit 2: Conditionals & Operators (3 lessons)

Students develop conditional logic in programs and plan larger creative projects that require decomposing actions into logical steps.

Objectives / Topics Covered	● Use if/then conditionals to control sprite behavior. ● Decompose larger programs into manageable steps. ● Plan multi-step animations using logic and structure.
Lessons	**Conditionals: Mazes**

	● Create a maze program that uses conditionals to navigate paths. **Plan a Quest (2 day lesson)** ● Break down and plan the steps needed to build a quest-style program.

Unit 3: Variables & Lists (3 lessons)

Students build interactive programs that respond dynamically by using variables and lists to control data like speed, pitch, and shopping items.

Objectives / Topics Covered	● Create and use variables in creative contexts. ● Use lists to store and manage multiple data points. ● Practice responsible computing by giving proper credit.
Lessons	**Variables in Dance** ● Use variables to change dance speed and pitch, creating a dynamic and musical animation. **Lists: Shopping with Scout (2 day lesson)** ● Build a shopping simulator that uses lists to track items and variables and operators to calculate prices. **Giving Credit Through Attributions** ● Give appropriate attribution when creating or remixing programs online.

Unit 4: Clones & Functions (3 lessons)

This unit focuses on game design and programming efficiency through the use of clones, object manipulation, and early object-oriented concepts like classes.

Objectives / Topics Covered	● Use clones to program repeated sprite behaviors in games. ● Understand classes and objects in a Scratch context. ● Use randomization and input to vary object characteristics.
Lessons	**Clones: Throwing Acorns Game** ● Create a game where acorns are cloned and thrown toward targets. **Clones in Games (2 day lesson)** ● Program an endless runner game using clones and explain their usefulness for repeating actions or objects. **Classes and Objects in Games (2 day lesson)** ● Learn about classes and objects while programming a game that randomizes object characteristics.

Unit 5: Culmination Projects (10 lessons)

In this unit, students use the design thinking process and programming concepts to build original projects and tell stories with data.

Objectives / Topics Covered	● Use programming skills to design a game from scratch. ● Apply design thinking to solve user-centered problems. ● Collect and present data using graphs and visuals.
Lessons	**Game Design Project (3 day lesson)** ● Create a custom game using loops, conditionals, variables, and other programming concepts. **Design an App (3 day lesson)** ● Use design thinking to plan an app that solves a real-world problem. **Inquiry Project: Line Graph (2 day lesson)**

	● Investigate a question and modify a program to present the results as a line graph. **Collaborating Globally (2 day lesson)** ● Create and share a program digitally, give and receive feedback, and practice collaboration by improving a project with input from peers.

Unit 8: Digital Literacy (9 lessons)

Students will explore how computing innovations, online collaboration, networks, and cybersecurity shape their digital lives. They'll build skills in productivity tools and learn how to research, organize, and present information responsibly.

Objectives / Topics Covered	● Understand how computing innovations and networks impact the world. ● Practice safe, responsible, and secure online behavior. ● Use collaboration and communication tools effectively. ● Organize, visualize, and present data using productivity software. ● Conduct and present research using digital tools.
Lessons	**Practicing with Computing Systems** ● Identify parts of a computing system and basic troubleshooting steps. **File Management and Data Exploration** ● Explain how digital data varies in size and where it can be stored. **Networks and Protocols** ● Learn how data is transferred through networks and compare WiFi, wired, and cellular connections. **Digital Identity** ● Understand how online actions impact digital identity and create a positive digital footprint. **Impacts of Computing: Innovation** ● Explore how computing innovations have changed the way we live and work. **Use and Search the Right Way** ● Search online effectively and provide proper attribution to sources. **Machine Learning: Teachable Machine** ● Train a machine learning model using Teachable Machine and explain how data quality impacts accuracy.

Idaho Computer Science 5th Grade Course Supplemental Materials

Resources	**Description**
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to computer science.
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.
All of these resources and more are found on the **Elementary Resources Page**.	