5th Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **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.

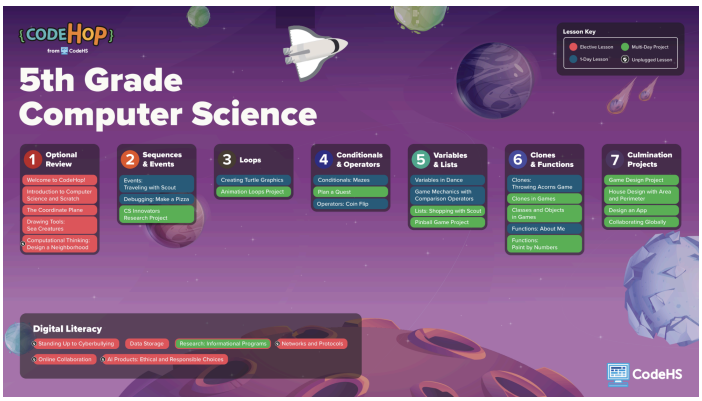
Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging 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. 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 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/20687/overview?lang=en



A clickable PDF can be found at https://codehs.com/K5CSRoadmaps.

Course Breakdown

Unit 1: Optional Review (4 weeks)

In this unit, students revisit key Scratch skills, build confidence using creative tools, and apply computational thinking in open-ended projects.

Objectives / Topics Covered	● Navigate the CodeHop Playground and Scratch editor. ● Create original projects using drawing tools and coordinate-based movement. ● Apply computational thinking skills to design solutions. ● Review foundational CS vocabulary and programming practices.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Log in and explore the CodeHop Playground as an introductory or standalone activity. **Introduction to Computer Science and Scratch** ● Define essential CS vocabulary and apply it in a simple Scratch program. **The Coordinate Plane** ● Animate sprites in Scratch using coordinate plane positioning for creative movement. **Drawing Tools: Sea Creatures** ● Design and program underwater characters using the full set of Scratch image editing tools. **Computational Thinking: Design a Neighborhood** ● Use decomposition, pattern recognition, and sequencing to design a digital neighborhood.

Unit 2: Sequences and Events (4 weeks)

In this unit, students explore how sequences and events control program flow, debug errors, and connect computing concepts to real-world innovations.

Objectives / Topics Covered	● Create programs using sequences and event-driven actions. ● Debug programs by identifying and fixing errors. ● Apply abstraction to summarize and present researched information. ● Explore real-world innovators in computer science.
Lessons	**Events: Traveling with Scout** ● Use event blocks to trigger actions between characters in a Scratch program. **Debugging: Make a Pizza** ● Break down a program into smaller parts and fix bugs to make the program function as intended. **CS Innovators Research Project (2-part lesson)** ● Research a computer science innovator and create an informational Scratch project using abstracted facts.

Unit 3: Loops (3 weeks)

Students explore how loops can simplify repetitive actions in programs through creative animations and art.

Objectives / Topics Covered	● Use repeat loops to simplify repeated actions. ● Apply loops to create animations with multiple scenes. ● Create geometric or artistic designs using loops and the pen tool. ● Understand how loops improve code efficiency.
Lessons	**Creating Turtle Graphics** ● Use the pen tool in Scratch to draw looping geometric designs inspired by turtle

	graphics. **Animation Loops Project (2-part lesson)** ● Program an animation with repeated scenes using repeat loop blocks.

Unit 4: Conditionals and Operators (4 weeks)

In this programming unit, students explore conditionals and operators to create responsive and interactive Scratch programs.

Objectives / Topics Covered	● Use if/then logic to control character behavior. ● Design algorithms that rely on decision-making. ● Apply operators like random and equality to introduce variation. ● Plan and decompose complex interactive projects.
Lessons	**Conditionals: Mazes** ● Create a program that uses conditionals to help characters navigate a maze. **Plan a Quest (2-part lesson)** ● Plan and break down the steps of an interactive quest program using conditionals and sequences. **Operators: Coin Flip** ● Use variables and operators to simulate a coin flip and display randomized outcomes.

Unit 5: Variables and Lists (7 weeks)

In this unit, students apply variables, lists, and comparison operators to create interactive programs with custom mechanics.

Objectives / Topics Covered	● Create and manipulate variables to control program behavior. ● Use lists to store and access multiple values. ● Apply comparison operators to enable conditional outcomes. ● Combine logic structures to design functional game mechanics.
Lessons	**Variables in Dance** ● Use variables to control the pitch and speed of dance moves in a rhythmic animation. **Game Mechanics with Comparison Operators** ● Combine comparison operators with variables to trigger game win or loss conditions. **Lists: Shopping with Scout (2-part lesson)** ● Build a shopping simulator using variables, lists, and operators to track items and simulate purchases. **Pinball Game Project (3-part lesson)** ● Design and develop a pinball-style game using key programming principles, including variables and lists.

Unit 6: Clones and Functions (8 weeks)

Students deepen their programming skills by using clones, custom functions, and object-based thinking to build advanced, interactive projects.

Objectives / Topics Covered	● Use clones to manage repeating elements in games. ● Create and apply functions with input values. ● Understand classes and objects in game development. ● Explore how randomization and input can customize behaviors.
Lessons	**Clones: Throwing Acorns Game** ● Build a game where characters throw acorns using clones to manage multiple objects. **Clones in Games (2-part lesson)** ● Program an endless runner game using clones and explain their usefulness in reducing

	repetitive code. **Classes and Objects in Games (2-part lesson)** ● Learn about object-oriented programming by creating interactive elements with randomized traits. **Functions: About Me** ● Design a personalized program that uses a function with input to display information. **Functions: Paint by Numbers (2-part lesson)** ● Create a visual program using functions and numerical input to control outcomes.

Unit 7: Culmination Projects (10 weeks)

In this unit, students apply their knowledge of programming, design, and problem-solving to build capstone projects that demonstrate creativity, real-world application, and collaboration.

Objectives / Topics Covered	● Integrate core programming concepts into original projects. ● Apply design thinking to solve real-world problems through apps or games. ● Explore how math and CS intersect through project-based learning. ● Collaborate and share ideas globally.
Lessons	**Game Design Project (3-part lesson)** ● Design and build a playable game using programming concepts such as loops, conditionals, and variables. **House Design with Area and Perimeter (2-part lesson)** ● Apply math and CS concepts by calculating area and perimeter to create a house layout using functions. **Design an App (3-part lesson)** ● Use the design thinking process to brainstorm, prototype, and program an app that meets user needs. **Collaborating Globally (2-part lesson)** ● *Lesson coming soon!*

Unit 8: Digital Literacy (7 weeks)

In this supplemental unit, students will explore how technology, data, and online tools influence our digital lives, focusing on ethical behavior, collaboration, and responsible use of emerging technologies.

Objectives / Topics Covered	● Identify and respond to cyberbullying and other online behaviors. ● Explain how data is stored and transferred across networks. ● Use research skills to develop informative digital content. ● Evaluate and reflect on ethical use of AI and digital collaboration tools.
Lessons	**Standing Up to Cyberbullying** ● Recognize hurtful online behaviors and describe appropriate ways to respond or take responsibility. **Data Storage** ● Explain how different types of data require varying amounts of storage based on detail and format. **Research: Informational Programs (2-part lesson)** ● Analyze sources and synthesize information into a Public Service Announcement (PSA) on healthy sleep habits. **Networks and Protocols** ● Compare WiFi, wired, and cellular networks and explain how protocols enable data transfer. **Online Collaboration** ● Understand how collaboration happens both online and offline, and how diverse perspectives improve group work. **AI Products: Ethical and Responsible Choices** ● Explore how AI products function and evaluate their benefits and challenges from

	different viewpoints.

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 **CodeHop Resources Page**.	