Idaho Computer Science 3rd Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Idaho Computer Science 3rd Grade Course introduces students to foundational programming concepts through 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 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.

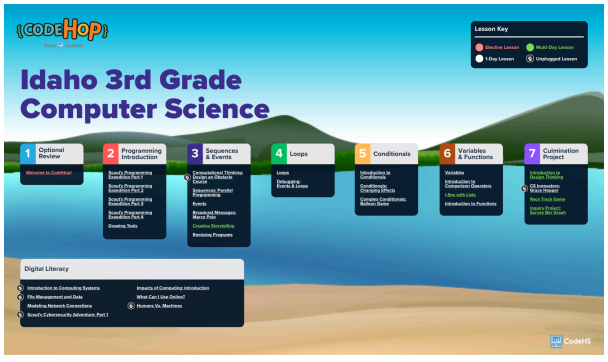
The course includes 36 lessons, each approximately 45 minutes long, providing a full year of instruction when taught once per week. While the course allows for instructional flexibility, some lessons are required to fully meet state computer science standards and are clearly identified within the syllabus.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

Programming Environment: Students will write and run programs that are saved in the CodeHop platform. The environment supports interactive, hands-on programming, enabling students to create and debug projects in a user-friendly interface.

Assessments: Some units conclude with a brief five-question quiz designed to assess student understanding of the concepts and skills introduced in the unit. Quizzes can be assigned digitally, completed as a QuizBuzz activity, or printed for paper-based completion.

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/26580/overview



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

Course Breakdown

Optional Review

This brief review lesson helps students get comfortable logging in and navigating the Playground.

Objectives / Topics Covered	● Log in and explore basic site features to prepare for upcoming lessons.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Practice logging in and exploring the Playground before starting a full lesson.

Unit 1: Programming Introduction (5 lessons)

In this story-driven unit, students follow Scout the Squirrel on an adventure through CodeHop while learning to animate characters, create stories, and use basic coding concepts.

Objectives / Topics Covered	● Explore the CodeHop interface and basic block types. ● Animate characters with motion, looks, and speech. ● Use sequences, events, and loops to build short stories. ● Customize characters and scenes with drawing tools.
Lessons	**Scout's Programming Expedition Part 1** ● Program a sprite to move and talk using basic commands. **Scout's Programming Expedition Part 2** ● Add sprites and create a sequence to animate part of the story. **Scout's Programming Expedition Part 3** ● Animate a Scout story using loops, events, and motion blocks. **Scout's Programming Expedition Part 4** ● Continue the animated story using events, looks, and motion for more interaction. **Drawing Tools (3-5.CT.1.3)** ● Customize sprites and backdrops using drawing tools. **Programming Introduction Quiz (5 questions)**

Unit 3: Sequences & Events (4 lessons)

Students will learn how to build programs that run multiple sequences at once, use events to trigger actions, and create interactive stories by combining these concepts creatively.

Objectives / Topics Covered	● Break down tasks using computational thinking. ● Build sequences and use events to trigger actions. ● Coordinate sprite interactions with broadcast messages. ● Remix programs responsibly and creatively.
Lessons	**Computational Thinking: Design an Obstacle Course (3-5.CT.5.7, 3-5.CT.5.8)** ● Create a program where multiple sequences run at the same time. **Sequences: Parallel Programming (3-5.CT.5.6)** ● Create a program where multiple sequences run at the same time. **Events** ● Use event blocks to start actions when keys are pressed or sprites are clicked. **Broadcast Messages: Marco Polo** ● Send and receive messages between sprites to coordinate actions. **Creative Storytelling (2 day lesson)** ● Plan and animate a story using sequences and event-based interactions. **Remixing Programs (3-5.CT.1.4)** ● Remix digital programs and give appropriate credit to original creators. **Sequences & Events Quiz (5 questions)**

Unit 4: Loops (3 lessons)

This unit introduces loops as a powerful tool to simplify repeated instructions and patterns. Students will also learn to debug programs involving loops and events.

Objectives / Topics Covered	● Use repeat loops to simplify repetitive code. ● Understand how loops improve efficiency and readability. ● Debug programs that include events and repetition.
Lessons	**Loops** ● Learn how loops repeat instructions and use them in a project. **Debugging: Events and Loops (3-5.CT.5.5)** ● Decompose a program to debug and make the program run as intended.

Unit 5: Conditionals (3 lessons)

Students explore conditional logic to make programs respond dynamically to different conditions and inputs.

Objectives / Topics Covered	● Understand and apply if/then and if/then/else logic. ● Use conditionals to build interactive and reactive programs. ● Differentiate between simple and complex conditionals.
Lessons	**Introduction to Conditionals** ● Learn what conditionals are and use if/then blocks in a program. **Conditionals: Changing Effects** ● Create a program using conditionals to trigger different visual responses. **Complex Conditionals: Balloon Game** ● Create a game using if/then/else blocks to respond to different outcomes. **Loops & Conditionals Quiz (5 questions)**

Unit 6: Variables & Functions (5 lessons)

In this unit, students use variables, lists, comparison operators, and custom functions to manage and organize data in more complex programs.

Objectives / Topics Covered	● Create and update variables. ● Use comparison operators in decision-making. ● Build lists to organize data for games and activities. ● Define and reuse functions to simplify large programs.
Lessons	**Variables** ● Learn what a variable is and how to create and change its value. **Introduction to Comparison Operators** ● Use comparison operators with numbers and variables in if/else logic. **I-Spy with Lists (2 classes)** ● Build an interactive I-Spy game using lists and variables to track items and responses. **Introduction to Functions** ● Learn how to define and use functions to simplify and organize code. **Variables, Lists & Functions Quiz (5 questions)**

Unit 8: Culmination Project (6 lessons)

Students apply their learning to creative projects that showcase design thinking, storytelling, data analysis, and game design using advanced programming concepts.

Objectives / Topics Covered	• Apply design thinking to create accessible solutions. • Explore computer science history and binary. • Design games using conditionals, variables, and lists. • Modify and present data using programs.
Lessons	**Introduction to Design Thinking (3-5.CT.2.3, 3-5.CT.5.1, 3-5.CT.5.2, 3-5.CT.5.3, 3-5.CT.5.4, 3-5.CT.5.9)** • Use the design thinking process to brainstorm, build, and improve a program. **CS Innovators: Grace Hopper (3-5.DL.3.3)** • Explain Grace Hopper's contributions and use binary code to solve puzzles. **Race Track Game (2 classes)** • Design a race car and track, then program a racing game that includes scorekeeping with conditionals, variables, and lists. **Inquiry Project: Survey Bar Graph (2 classes | 3-5.CT.2.1, 3-5.CT.2.4, 3-5.CT.2.5, 3-5.CT.2.6, 3-5.CT.3.3, 3-5.DL.3.2)** • Follow the inquiry process to investigate a topic and modify a program to display survey results using a bar graph.

Unit 9: Digital Literacy (7 lessons)

Students explore responsible technology use, systems thinking, data storage, cybersecurity, and the relationship between humans and machines.

Objectives / Topics Covered	• Understand the components and functions of computing systems. • Explore data storage, file types, and digital safety. • Model how data travels across networks. • Compare human and machine abilities in computing contexts.
Lessons	**Introduction to Computing Systems (3-5.CT.1.1, 3-5.DL.3.1)** • Identify parts of the computing system and solve basic hardware/software problems. **File Management and Data (3-5.CT.2.2)** • Explore how different types of digital data take up varying amounts of space and where that data can be stored. **Modeling Network Connections (3-5.CT.4.1)** • Show how data travels between devices using a network. **Scout's Cybersecurity Adventure: Part 1 (3-5.CT.3.2, 3-5.DL.2.1, 3-5.DL.2.2)** • Identify common cyber threats and strategies for staying safe online. **Impacts of Computing: Introduction (3-5.DL.4.1)** • Explore how computing has changed communities and evolved over time. **What Can I Use Online? (3-5.CT.3.1, 3-5.CT.3.4, 3-5.DL.1.1)** • Practice safe online research, use keywords effectively, and cite sources properly. **Humans vs. Machines (3-5.CT.1.2)** • Compare human and computer performance, and describe computer perception and limitations. **Digital Literacy Quiz (5 questions)**

Idaho Computer Science 3rd 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 Templates	Empower your students to explore and express their knowledge creatively with our versatile 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 **CodeHop Resources Page**.	