3rd Grade Computer Science Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The 3rd Grade Computer Science 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.

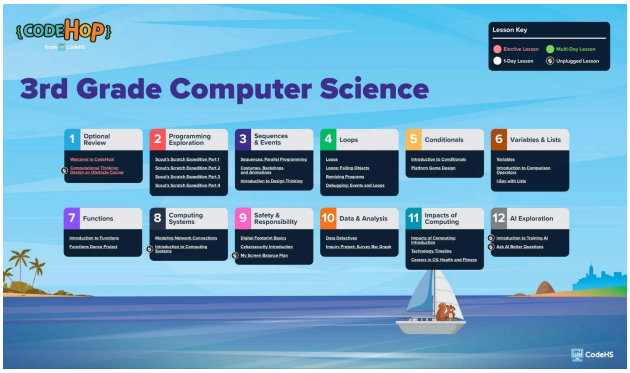
The course includes 36 lessons, each approximately 45 minutes long, providing a full year of instruction when taught once per week. While the course allows for instructional flexibility, some lessons are required to fully meet CSTA computer science standards and are clearly identified within the syllabus.

Standards Alignment Note: Lessons that list standards in parentheses next to the title are required to fully meet CSTA computer science standards. Lessons without a standards tag support spiral review, practice, or enrichment.

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/28415/overview



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

Course Breakdown

Unit 1: Optional Review (2 lessons)

These optional lessons are available for review at the start of 3rd grade. They are not counted toward the 36-lesson course total.

Objectives / Topics Covered	● Review foundational CodeHop navigation and programming concepts. ● Build confidence before beginning new 3rd grade content.
Lessons	**Welcome to CodeHop!** ● Learn how to log in and use the CodeHop Playground. This short introductory lesson can be used on its own, or right before a full lesson. **Computational Thinking: Design an Obstacle Course** ● Use computational thinking to design a school.

Unit 2: Programming Exploration (4 lessons)

In this module, students transition into the programming environment through the Scout's Programming Expedition series, exploring expanded features across four progressive lessons.

Objectives / Topics Covered	● Navigate the CodeHop programming environment to create and run programs. ● Apply sequencing and algorithmic thinking in a new programming platform. ● Build increasingly complex programs across a multi-part series.
Lessons	**Scout's Programming Expedition Part 1** (E3-ALG-PS-01) ● Use basic programming commands to program a sprite to move and talk. **Scout's Programming Expedition Part 2** (E3-ALG-PS-01) ● Add sprites and create a sequence to animate a story. **Scout's Programming Expedition Part 3** (E3-ALG-PS-01) ● Create an animated Scout story using loops, events, looks, and motion blocks. **Scout's Programming Expedition Part 4** (E3-ALG-PS-01) ● Create an animated Scout story using events, looks, and motion blocks.

Unit 3: Sequences & Events (3 lessons)

In this module, students deepen their understanding of sequences and events in Programming, working with parallel programming, character animations, and design thinking.

Objectives / Topics Covered	● Use parallel sequences to have multiple sprites act simultaneously. ● Apply costume and backdrop changes to create animations. ● Use design thinking to plan and create a purposeful program.
Lessons	**Sequences: Parallel Programming** ● Create a program using parallel sequences. **Costumes, Backdrops, and Animations** ● Create a program that includes animated sprites and interactive backdrops. **Introduction to Design Thinking** (E3-PRO-PD-05, E3-SOC-HU-18) ● Use the design thinking process to make a tool accessible to more users.

Unit 4: Loops (4 lessons)

In this module, students apply loops in increasingly complex programs, creating multi-class projects and practicing debugging loop-based code.

Objectives / Topics Covered	● Use repeat loops to automate sequences and reduce repeated code. ● Apply loops to build a multi-class game project. ● Remix an existing program to add new features and creativity. ● Debug programs that combine events and loops.
Lessons	**Loops** ● Explain that a loop repeats one or more instructions, and use loops in CodeHop. **Loops: Falling Objects** (2 classes | E3-ALG-PS-01) ● Create a program using different loops and compare the advantages and disadvantages of each loop algorithm. **Remixing Programs** (E3-PRO-PD-04) ● Remix digital programs and give appropriate credit to original creators. **Debugging: Events and Loops** (E3-PRO-RD-07, E3-PRO-TR-08) ● Decompose a program to debug and make the program run as intended.

Unit 5: Conditionals (2 lessons)

In this module, students learn how conditional statements allow programs to make decisions, culminating in a complete platform game.

Objectives / Topics Covered	● Use if/then conditional blocks to make programs respond to different inputs. ● Design and build a complete platform game using all concepts learned so far.
Lessons	**Introduction to Conditionals** (E3-ALG-PS-01, E3-PRO-PD-04) ● Explain what a conditional is in programming and create a program that uses if/then blocks. **Platform Game Design** (2 classes | E3-ALG-PS-01, E3-ALG-IM-03, E3-PRO-PD-04, E3-PRO-RD-07, E3-PRO-TR-08) ● Implement the mechanics of a platform game for a player to navigate, use keyboard events to add player movement, and use a forever loop to check and respond to game conditions. Students will debug and improve their programs.

Unit 6: Variables & Lists (3 lessons)

In this module, students learn how variables store and update data in programs, and how lists manage collections of information.

Objectives / Topics Covered	● Define and use variables to store and update information in a program. ● Use comparison operators to evaluate conditions in programs. ● Use lists to store and access collections of data.
Lessons	**Variables** (E3-PRO-VD-06) ● Explain what a variable is and create and change the value of a variable in a program. **Introduction to Comparison Operators** (E3-PRO-VD-06) ● Use comparison operators as the condition in an if/else block, and use numbers and variables as values in a comparison operator. **I-Spy with Lists** (2 classes | E3-PRO-VD-06) ● Create an I-Spy style activity using variables and lists.

Unit 7: Functions (2 lessons)

In this module, students are introduced to functions as reusable blocks of code that organize programs into named, callable sections.

Objectives / Topics Covered	● Define what a function is and explain how it organizes reusable code. ● Create and call custom functions. ● Apply functions to build a structured, creative programming project.
Lessons	**Introduction to Functions** ● Create and use functions in a program. **Functions Dance Project** (3 classes) ● Create and use functions to call dance moves in a sequence that aligns with the music.

Unit 8: Computing Systems (2 lessons)

In this module, students explore how computing systems are connected and how networks enable data to travel between devices.

Objectives / Topics Covered	● Model how devices connect and communicate through a network. ● Identify and describe the components of a computing system including hardware and software.
Lessons	**Modeling Network Connections** (E3-SYS-NT-14) ● Describe how messages move between devices using a network and create a program that shows how data travels from one device to another. **Introduction to Computing Systems (Unplugged)** (E3-SYS-HW-12) ● Identify parts of the computing system and solve simple hardware and software problems.

Unit 9: Safety & Responsibility (3 lessons)

In this module, students explore digital safety, cybersecurity, and screen balance through research-based activities and reflective planning.

Objectives / Topics Covered	● Understand what a digital footprint is and how to manage it responsibly. ● Identify cybersecurity threats and strategies for staying safe online. ● Create a personal plan for healthy and balanced screen time.
Lessons	**Digital Footprint Basics** (E3-SYS-SE-13) ● Describe safe and unsafe online behaviors and explain how those choices affect a digital footprint. **Cybersecurity Introduction** (E3-SYS-SE-13) ● Learn about cybersecurity, recognize online threats, and identify how to stay safe online. **My Screen Balance Plan (Unplugged)** (E3-SOC-ET-17) ● Define screen time and balance, develop healthy technology routines, and write an opinion piece on the most important rule for health and safety.

Unit 10: Data & Analysis (2 lessons)

In this module, students collect, organize, and analyze data to draw conclusions, applying skills in a multi-class inquiry project.

Objectives / Topics Covered	● Collect and organize data to identify patterns and draw conclusions. ● Create visual representations of data to communicate findings.
Lessons	**Data Detectives** (E3-DAT-DC-09, E3-DAT-DI-10) ● Evaluate data for reliability and then analyze the data to draw conclusions and make predictions. **Inquiry Project: Survey Bar Graph** (2 classes | E3-DAT-DI-10, E3-DAT-IM-11) ● Evaluate data for reliability and then analyze the data to draw conclusions and make predictions.

Unit 11: Impacts of Computing (3 lessons)

In this module, students examine how computing affects society, explore technology history, and discover CS careers in health and fitness.

Objectives / Topics Covered	● Describe how computing has impacted daily life and society over time. ● Trace the history of key technology milestones. ● Explore how computer science careers solve real-world challenges.
Lessons	**Impacts of Computing: Introduction** (E3-SYS-IM-15, E3-SOC-HI-16) ● Explain how technology and culture influence each other and create a project showing a past technology, its present version, and an improvement idea. **Technology Timeline** (E3-SOC-HI-16, E3-SOC-HU-18) ● Create an interactive timeline to illustrate the key developments in music player technology and explain how music player technology has influenced cultural practices. **Careers in CS: Health and Fitness** (E3-SOC-CE-19) ● Explain how fitness and coding can amplify human strengths and create an animation to demonstrate how technology can be used in health and fitness.

Unit 12: AI Exploration (2 lessons)

In this module, students explore how AI systems are trained to recognize patterns and practice crafting better questions for AI tools.

Objectives / Topics Covered	● Explain how AI systems are trained using labeled data to recognize patterns. ● Craft effective prompts and questions to get useful results from AI tools.
Lessons	**Introduction to Training AI (Unplugged)** (E3-ALG-ML-02) ● Observe and explain how an AI system's decisions change when it is given more data. **Ask AI Better Questions (Unplugged)** ● Write clear prompts for AI and decide if the responses are reasonable.

3rd 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 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.
These resources and more are found on the **CodeHop Resources Page**.	