



# New York Computer Science and Digital Fluency: 3rd Grade Course Syllabus

One Year for Elementary School, 36 Hours

## Course Overview and Goals

The **New York Computer Science and Digital Fluency: 3rd Grade** 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. 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 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/NY\\_3/overview](https://codehs.com/course/NY_3/overview)



A clickable PDF can be found at <https://codehs.com/NY-K-5Roadmaps>

## Course Breakdown

### Unit 1: Getting Started (3 lessons)

Students are introduced to the CodeHop environment, computing systems, and foundational computer science vocabulary while applying computational thinking to a design challenge.

Objectives / Topics Covered	<ul style="list-style-type: none"><li>● Log in and navigate the CodeHop Playground</li><li>● Identify parts of a computing system and troubleshoot basic problems</li><li>● Apply computational thinking to design a sequence of steps</li><li>● Define key computer science vocabulary and create a simple program</li></ul>
Lessons	<p><b>Welcome to CodeHop!</b></p> <ul style="list-style-type: none"><li>● Log in and explore the CodeHop Playground for the first time.</li></ul> <p><b>Introduction to Computing Systems (2-3.IC.6, 2-3.NSD.1, 2-3.NSD.2, 2-3.NSD.3)</b></p> <ul style="list-style-type: none"><li>● Identify parts of a computing system and solve simple hardware and software problems.</li></ul> <p><b>Computational Thinking: Design an Obstacle Course (2-3.CT.1)</b></p> <ul style="list-style-type: none"><li>● Use computational thinking concepts to plan and design an obstacle course.</li></ul> <p><b>Introduction to Computer Science (2-3.CT.5, 2-3.DL.4)</b></p> <ul style="list-style-type: none"><li>● Define key computer science vocabulary and write a first program in CodeHop.</li></ul>

### Unit 2: Programming Exploration (4 lessons)

Students follow Scout through a four-part story-driven series, building programming skills by adding sprites, sequences, loops, events, and motion blocks to animate an original story.

Objectives / Topics Covered	<ul style="list-style-type: none"><li>● Use basic commands to make a sprite move and speak</li><li>● Add sprites and create sequences to animate a story</li><li>● Combine loops, events, and motion blocks in a program</li><li>● Build and extend a multi-part animated story project</li></ul>
Lessons	<p><b>Scout's Programming Expedition: Part 1 (2-3.CT.9)</b></p> <ul style="list-style-type: none"><li>● Use basic programming commands to make a sprite move and talk.</li></ul> <p><b>Scout's Programming Expedition: Part 2</b></p> <ul style="list-style-type: none"><li>● Add sprites and build a sequence to animate a scene from Scout's story.</li></ul> <p><b>Scout's Programming Expedition: Part 3</b></p> <ul style="list-style-type: none"><li>● Animate Scout's story using loops, events, looks, and motion blocks together.</li></ul> <p><b>Scout's Programming Expedition: Part 4</b></p> <ul style="list-style-type: none"><li>● Continue Scout's story using events, looks, and motion blocks in a more complex animation.</li></ul>

### Unit 3: Sequences and Events (4 lessons)

Students use drawing tools to create custom sprites and backdrops, then build programs using parallel sequences, events, and broadcast messages.

Objectives / Topics Covered	<ul style="list-style-type: none"><li>● Create custom sprites and backdrops with drawing tools</li><li>● Build parallel sequences to run multiple scripts at once</li><li>● Use event blocks to trigger actions in a program</li><li>● Use broadcast messages to make sprites interact</li></ul>
Lessons	<p><b>Drawing Tools (2-3.CT.9, 2-3.DL.4)</b></p> <ul style="list-style-type: none"><li>● Create customized sprites and backdrops using CodeHop's drawing tools.</li></ul>

	<p><b>Sequences: Parallel Programming (2-3.CT.9)</b></p> <ul style="list-style-type: none"> <li>• Build a program where multiple sequences run in parallel at the same time.</li> </ul> <p><b>Events (2-3.DL.4)</b></p> <ul style="list-style-type: none"> <li>• Create a program that uses event blocks to trigger different actions.</li> </ul> <p><b>Broadcast Messages: Marco Polo (2-3.CT.9, 2-3.DL.4)</b></p> <ul style="list-style-type: none"> <li>• Use broadcast messages to trigger actions between sprites in a program.</li> </ul>
--	---

#### Unit 4: Loops (6 lessons)

Students explore different types of loops, compare loop algorithms, remix existing programs with attribution, and apply loops in creative projects about health, fitness, and cultural celebrations.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>• Explain how loops work and use them in a program</li> <li>• Compare different loop types and their trade-offs</li> <li>• Remix a program and give credit to the original creator</li> <li>• Apply loops to build animations with real-world themes</li> </ul>
Lessons	<p><b>Loops (2-3.CT.4, 2-3.CT.9, 2-3.DL.4)</b></p> <ul style="list-style-type: none"> <li>• Explain that a loop repeats instructions and use loops to simplify a program in CodeHop.</li> </ul> <p><b>Loops: Falling Objects (2 classes   2-3.CT.4, 2-3.CT.6, 2-3.CT.9, 2-3.DL.4)</b></p> <ul style="list-style-type: none"> <li>• Create a program using different loop types and compare the advantages of each approach.</li> </ul> <p><b>Remixing Programs (2-3.DL.7, 2-3.IC.2)</b></p> <ul style="list-style-type: none"> <li>• Remix an existing digital program and give proper credit to the original creator.</li> </ul> <p><b>Careers in CS: Health and Fitness (2-3.IC.7)</b></p> <ul style="list-style-type: none"> <li>• Explain how coding supports health and fitness and create an animation showing technology in action.</li> </ul> <p><b>Winter Celebrations Around the World (2-3.CT.5, 2-3.DL.4)</b></p> <ul style="list-style-type: none"> <li>• Research a winter celebration and create a program that shares how that holiday is celebrated.</li> </ul>

#### Unit 5: Conditionals (2 lessons)

Students learn what conditionals are and how if/then logic controls program behavior, then apply conditionals to create programs that respond to user input and change visual effects.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>• Explain what a conditional is and how if/then logic works</li> <li>• Create a program that uses if/then blocks</li> <li>• Apply conditionals to change effects in a program</li> <li>• Distinguish between programs that do and don't use conditionals</li> </ul>
Lessons	<p><b>Introduction to Conditionals (2-3.DL.4)</b></p> <ul style="list-style-type: none"> <li>• Explain what a conditional is and build a program that uses if/then blocks.</li> </ul> <p><b>Conditionals: Changing Effects (2-3.CT.8, 2-3.DL.4)</b></p> <ul style="list-style-type: none"> <li>• Create a program that uses conditionals to change sprite effects based on conditions.</li> </ul>

#### Unit 6: Variables (4 lessons)

Students learn how variables store and change data, use comparison operators in conditionals, and complete an inquiry project that collects and visualizes survey data.

Objectives	<ul style="list-style-type: none"> <li>• Explain what a variable is and change its value in a program</li> </ul>
------------	--

/ Topics Covered	<ul style="list-style-type: none"> <li>● Use comparison operators in if/else blocks</li> <li>● Combine variables and conditionals in a program</li> <li>● Follow the inquiry process to collect and display data</li> </ul>
Lessons	<p><b>Variables (2-3.CT.7, 2-3.DL.4)</b></p> <ul style="list-style-type: none"> <li>● Define a variable, set its value, and update it inside a program.</li> </ul> <p><b>Introduction to Comparison Operators (2-3.CT.7, 2-3.CT.8, 2-3.DL.4)</b></p> <ul style="list-style-type: none"> <li>● Use comparison operators as conditions in if/else blocks with numbers and variables.</li> </ul> <p><b>Inquiry Project: Survey Bar Graph (2 classes   2-3.CT.2, 2-3.CT.3, 2-3.DL.4, 2-3.NSD.4)</b></p> <ul style="list-style-type: none"> <li>● Follow the inquiry process, collect survey data, and modify a program to display the results as a bar graph.</li> </ul>

### Unit 7: Functions (4 lessons)

Students learn how to define and call functions to organize code into reusable blocks, then apply functions creatively in a dance project synchronized to music.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Explain what a function is and why functions are useful</li> <li>● Create and call a function in a program</li> <li>● Use functions to organize a complex sequence of actions</li> <li>● Apply functions to build a dance animation synced to music</li> </ul>
Lessons	<p><b>Introduction to Functions (2-3.CT.9, 2-3.DL.4)</b></p> <ul style="list-style-type: none"> <li>● Create and call functions to organize repeated actions into reusable blocks of code.</li> </ul> <p><b>Functions Dance Project (3 classes   2-3.CT.9, 2-3.DL.4)</b></p> <ul style="list-style-type: none"> <li>● Use functions to define and sequence dance moves in an animation that aligns with a music track.</li> </ul>

### Unit 8: Culmination Project (2 lessons)

Students apply the design thinking process to create an accessible tool, synthesizing their programming skills and user-centered thinking into a final project.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Describe the stages of the design thinking process</li> <li>● Identify a user need and design a solution</li> <li>● Build and test a program using design thinking</li> <li>● Reflect on and present a completed design project</li> </ul>
Lessons	<p><b>Introduction to Design Thinking (2 classes   2-3.CT.10, 2-3.CT.9, 2-3.DL.2, 2-3.DL.4)</b></p> <ul style="list-style-type: none"> <li>● Use the design thinking process to redesign a tool so it is more accessible to a wider group of users.</li> </ul>

### Unit 9: Digital Literacy (7 lessons)

Students explore how technology shapes culture, practice digital etiquette and online safety, learn basic cryptography, conduct research with attribution, and investigate how machines learn.

Objectives / Topics Covered	<ul style="list-style-type: none"> <li>● Explain how technology and culture influence each other</li> <li>● Demonstrate safe and responsible online communication</li> <li>● Describe how digital footprints and cybersecurity tools protect information</li> <li>● Conduct online research with proper source attribution</li> </ul>
-----------------------------	---

Lessons	<p><b>Impacts of Computing: Introduction (2-3.IC.1, 2-3.IC.3, 2-3.IC.6)</b></p> <ul style="list-style-type: none"> <li>Explain how technology and culture influence each other and create a project comparing a past and present technology.</li> </ul> <p><b>Introduction to Digital Etiquette and Communication (2-3.DL.6, 2-3.DL.7)</b></p> <ul style="list-style-type: none"> <li>Demonstrate proper digital etiquette when communicating in an online community.</li> </ul> <p><b>Online Risks &amp; Protection (2-3.CY.2, 2-3.CY.3, 2-3.CY.5, 2-3.IC.4)</b></p> <ul style="list-style-type: none"> <li>Practice secure online habits and identify tools that protect personal information.</li> </ul> <p><b>Digital Footprint Basics (2-3.CY.1, 2-3.IC.4)</b></p> <ul style="list-style-type: none"> <li>Describe safe and unsafe online behaviors and explain how they affect a digital footprint.</li> </ul> <p><b>Introduction to Cryptography (2-3.CY.4)</b></p> <ul style="list-style-type: none"> <li>Learn basic cryptography concepts and solve multiple cipher challenges.</li> </ul> <p><b>What Can I Use Online? (2-3.DL.3)</b></p> <ul style="list-style-type: none"> <li>Search for information online to answer a question and provide proper attribution to sources.</li> </ul> <p><b>How Machines Learn (2-3.CT.2, 2-3.IC.1, 2-3.IC.5)</b></p> <ul style="list-style-type: none"> <li>Explain different machine learning approaches and modify a program to model how AI makes predictions.</li> </ul>
---------	--

## New York Computer Science and Digital Fluency: 3rd Grade Supplemental Materials

Resources	Description
<a href="#">Parent Welcome Letter (Spanish)</a>	Send this letter home to introduce families to their new computer science curriculum.
<a href="#">Warm-Up Activities</a>	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.
<a href="#">Program Self-Assessment (Spanish)</a>	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.
<a href="#">Peer Review Resources (Spanish)</a>	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.
<a href="#">Lesson Reflection &amp; Computational Thinking (Spanish)</a>	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 <a href="#">CodeHop Resources Page</a> .	