New York Computer Science and Digital Fluency: 5th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **New York Computer Science and Digital Fluency: 5th 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.

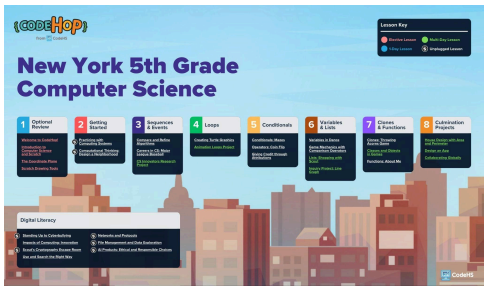
Learning Environment: This course is teacher-led and includes ready-to-use lessons following a consistent structure: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Instruction follows an "I do, we do, you do" model and incorporates spiral review to reinforce concepts 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_5/overview



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

Course Breakdown

Optional Review

Students review foundational CodeHop skills from 4th grade — the interface, CS vocabulary, coordinate plane, and drawing tools — to refresh their programming foundation before new content.

Objectives / Topics Covered	• Log in and navigate the CodeHop Playground • Define key computer science vocabulary and write a simple program • Use the coordinate plane to position and animate sprites • Create custom sprites and backdrops with the drawing tools
Lessons	**Welcome to CodeHop!** • Log in and explore the CodeHop Playground interface. **Introduction to Computer Science (4-6.DL.4)** • Define important computer science vocabulary and create a simple program. **The Coordinate Plane (4-6.DL.4)** • Create an open-ended animation using the coordinate plane to position sprites. **Drawing Tools (4-6.DL.4)** • Create customized sprites and backdrops using CodeHop's drawing tools.

Unit 1: Getting Started (2 lessons)

Students identify and troubleshoot parts of a computing system, then apply computational thinking to a neighborhood design challenge that requires sequencing and decomposition.

Objectives / Topics Covered	• Identify hardware and software components of a computing system • Troubleshoot simple hardware and software problems • Apply computational thinking to plan a multi-step design • Decompose a complex problem into manageable steps
Lessons	**Practicing with Computing Systems (4-6.CT.1, 4-6.DL.5, 4-6.NSD.2, 4-6.NSD.3)** • Identify parts of a computing system and diagnose simple hardware and software problems. **Computational Thinking: Design a Neighborhood (4-6.CT.10, 4-6.CT.4)** • Use computational thinking strategies to design a neighborhood through sequencing and abstraction.

Unit 2: Sequences and Events (3 lessons)

Students compare and refine algorithms for efficiency, then research a CS innovator and translate their findings into an informational program.

Objectives / Topics Covered	• Compare multiple algorithms for the same task and choose the most efficient • Refine an algorithm based on evaluation criteria • Research a topic and abstract key facts from a source • Communicate research findings through an interactive program
Lessons	**Compare and Refine Algorithms (4-6.CT.6, 4-6.DL.4)** • Compare multiple algorithms for the same task and determine which is most appropriate and efficient. **CS Innovators Research Project (2 classes | 4-6.IC.5, 4-6.IC.7)** • Research a CS innovator, extract key facts from an article, and present findings as an informational program.

Unit 3: Loops (2 lessons)

Students use the pen tool to create geometric turtle graphics with loops, then analyze and improve a mobile app game by modifying existing code and adding new features.

Objectives / Topics Covered	● Use the pen tool with loops to draw turtle graphics patterns ● Analyze existing code to understand how a program works ● Modify code to improve or extend a program's functionality ● Connect coding skills to real-world mobile app development
Lessons	**Creating Turtle Graphics (4-6.CT.4, 4-6.DL.4)** ● Use the pen tool and loops to create repeating geometric turtle graphics patterns. **Careers in CS: Mobile Apps (4-6.IC.7)** ● Analyze and improve a game by modifying existing code and adding new functionality.

Unit 4: Conditionals (3 lessons)

Students use pseudocode and conditionals to build a maze program, apply operators in a coin flip simulator, and practice giving proper attribution when remixing programs.

Objectives / Topics Covered	● Use pseudocode to plan a program before coding it ● Create a program that uses conditionals to respond to input ● Use operators and variables in combination with conditionals ● Give proper attribution when creating or remixing programs
Lessons	**Conditionals: Mazes (4-6.CT.8, 4-6.DL.4)** ● Write pseudocode to plan a maze program, then build it using conditional statements. **Operators: Coin Flip (4-6.CT.4, 4-6.CT.8, 4-6.DL.4)** ● Create a coin flip simulator using variables and operators to produce random outcomes. **Giving Credit Through Attributions (4-6.IC.2)** ● Practice giving appropriate attribution when creating or remixing programs online.

Unit 5: Variables and Lists (6 lessons)

Students use variables to control program behavior, apply comparison operators for game mechanics, manage data with lists, and complete an inquiry project that visualizes results as a line graph.

Objectives / Topics Covered	● Use variables to control pitch and animation speed in a program ● Apply comparison operators and variables to build game-ending mechanics ● Use lists to store and retrieve multiple values in a program ● Follow the inquiry process to collect and display data as a line graph
Lessons	**Variables in Dance (4-6.CT.7, 4-6.DL.4)** ● Use variables to control pitch and dance speeds in an animated program. **Game Mechanics with Comparison Operators (4-6.CT.4, 4-6.CT.8, 4-6.CT.9, 4-6.DL.4)** ● Use comparison operators and variables to create win/lose ending mechanics in a game. **Lists: Shopping with Scout (2 classes)** ● Build a shopping simulator that uses variables, lists, and operators to manage inventory. **Inquiry Project: Line Graph (2 classes | 4-6.CT.1, 4-6.CT.2, 4-6.CT.3)** ● Follow the inquiry process, collect data, and modify a program to display results as a line graph.

Unit 6: Clones and Functions (3 lessons)

Students build a clones-based throwing game, then explore classes and objects by creating an interactive game that uses randomizers to vary object characteristics.

Objectives / Topics Covered	● Use clones to generate multiple sprite copies in a game ● Control clone behavior with variables and events ● Explain the concepts of classes and objects in programming ● Use randomizers to vary object characteristics in a game
Lessons	**Clones: Throwing Acorns Game (4-6.CT.7, 4-6.CT.8, 4-6.DL.4)** ● Create a throwing acorns game that uses clones to generate multiple projectiles. **Classes and Objects in Games (2 classes | 4-6.CT.10, 4-6.CT.4, 4-6.CT.7, 4-6.CT.8, 4-6.DL.4)** ● Learn about classes and objects while building an interactive game that uses randomizers to change object characteristics.

Unit 7: Culmination Projects (7 lessons)

Students apply functions to a house design project, use design thinking to plan an original app, and collaborate globally to create a digital tourism advertisement.

Objectives / Topics Covered	● Use functions with parameters to build a geometric house design ● Apply the design thinking process to plan an app for a user need ● Collaborate with others digitally to co-create a program ● Present and reflect on a finished project
Lessons	**House Design with Area and Perimeter (2 classes | 4-6.CT.1, 4-6.CT.4, 4-6.CT.5, 4-6.CT.7, 4-6.CT.8, 4-6.DL.4)** ● Calculate area and perimeter to create a room-based house design using functions. **Design an App (3 classes | 4-6.CT.10, 4-6.CT.4, 4-6.CT.8, 4-6.CT.9, 4-6.DL.4, 4-6.IC.6, 4-6.NSD.1)** ● Use the design thinking process to plan and prototype an app that solves a real user need. **Collaborating Globally (2 classes | 4-6.DL.2)** ● Collaborate with students from another community to create a shared digital tourism advertisement.

Unit 8: Digital Literacy (10 lessons)

Students examine cyberbullying, digital reputation, cybersecurity, cryptography, computing's cultural impacts, online research, networking, file management, and the ethics of AI products.

Objectives / Topics Covered	● Recognize types of cyberbullying and describe how to respond as an upstander ● Explain how online actions shape digital reputation and identity ● Model how networks transfer data using rules and packets ● Analyze the benefits and challenges of AI products from multiple perspectives
Lessons	**Reporting Cyberbullying (4-6.DL.6, 4-6.DL.7)** ● Recognize types of cyberbullying, explain how interactions escalate, and demonstrate how to report and act as an upstander. **Digital Reputation and Identity (4-6.CY.2, 4-6.CY.3, 4-6.DL.6, 4-6.DL.7, 4-6.IC.4)** ● Analyze how online actions affect digital reputation and explain strategies to protect personal data and digital security.

	Cybersecurity & Privacy (2 classes | 4-6.CY.1, 4-6.CY.2, 4-6.CY.3, 4-6.CY.5, 4-6.IC.4) ● Create an animated program that demonstrates a solution to a real-world cybersecurity issue. **Impacts of Computing: Innovation (4-6.IC.1, 4-6.IC.3)** ● Compare past and present versions of a technology, explain its cultural impacts, and predict how emerging technologies may affect communities. **Cryptography Escape Room (4-6.CY.4)** ● Create and solve multiple cipher challenges to demonstrate understanding of basic cryptography. **Use and Search the Right Way (4-6.DL.3, 4-6.IC.2)** ● Search for information online to answer a question and provide proper attribution to sources. **Network Connections (4-6.NSD.4)** ● Compare wired and wireless network types and model how devices communicate and transfer data using rules and packets. **File Management and Data Exploration (4-6.CT.7, 4-6.NSD.5)** ● Explain how different types of digital data take up varying storage space and identify where that data can be stored. **AI Products: Ethical and Responsible Choices (4-6.CY.1, 4-6.CY.2, 4-6.DL.7, 4-6.IC.1, 4-6.IC.3, 4-6.IC.5)** ● Describe how AI products work and analyze their benefits and challenges from multiple perspectives.

New York Computer Science and Digital Fluency: 5th Grade 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.
These resources and more are found on the **CodeHop Resources Page**.	