Nova Scotia Information and Communication Technology 2nd Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Nova Scotia Information and Communication Technology 2nd Grade Course** introduces students to foundational programming concepts through **CodeHopJr**, 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 **contact hours**, with each lesson approximately 45 minutes long. This provides a full school year of material if teaching one lesson per week.

Programming Environment: Students will write and run programs in **CodeHopJr** 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/NS_2/overview



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

Course Breakdown

Optional Review

This optional review unit is designed to support students who need more time exploring CodeHopJr or who would benefit from additional practice before or after completing core lessons.

Objectives / Topics Covered	● Log in and navigate the Playground. ● Reinforce understanding of basic programming concepts such as events, loops, and message blocks.
Lessons	**Welcome to CodeHop!** ● Learn how to log in and use the Playground; a quick intro lesson to get started with the platform. **Introduction to Programming** ● Explore the CodeHopJr interface and create a scene by adding and placing characters. **Events** ● Learn what an event is in programming and use event blocks to trigger actions in a project. **Introduction to Repeat Loops** ● Use repeat loops to run a section of code multiple times and simplify repeated actions. **Forever Loop Dance Party** ● Create a fun animation where characters repeat movements continuously using the “repeat forever” loop. **Introduction to Message Events** ● Build a relay-style animation where characters interact using message blocks to control the flow of the program.

Unit 1: Getting Started (3 weeks)

In this unit, students will explore the basics of computing by learning the functions of computers and their components, practice identifying positive and negative online behaviors, and apply foundational computational thinking skills like pattern recognition, sequencing, and task decomposition to real-life routines.

Objectives / Topics Covered	● Understand the basic components of a computer. ● Learn responsible ways to use technology. ● Apply computational thinking in everyday routines.
Lessons	**Computer Basics: Connections (2.CCT.a.1, 2.CCT.a.2)** ● Learn what a computer is, how we use it, and what to do when it doesn’t work. **Practicing Responsible Technology Use (2.DC.a.1, 2.DC.a.2, 2.DC.b.1)** ● Explore ways to use technology safely, respectfully, and responsibly in daily life. **Computational Thinking: School Day Routines** ● Use computational thinking to find patterns, break down routines, and organize steps to better understand everyday school activities.

Unit 2: Sequences and Events (10 weeks)

In the *Sequences and Events* module, students build foundational programming skills by creating and adjusting algorithms, using the grid to control movement, and designing interactive programs with events. They also develop problem-solving abilities by identifying and fixing errors in code through debugging practice.

Objectives / Topics	● Develop simple algorithms to control character movement and behavior. ● Use events to trigger actions.

Covered	● Apply grid-based positioning to guide character actions. ● Identify and correct errors in sequences and events.
Lessons	**Drawing Tools: Farm Scene (2.DT.a.1, 2.DT.c.1)** ● Use drawing tools to customize characters and backgrounds. **Debugging: Events and Sequences (2.DT.a.1, 2.CCT.b.3, 2.CCT.c.2)** ● Practice finding and fixing errors in code to make a program work as expected. **Introduction to the Grid (2.DT.a.1, 2.CCT.b.3)** ● Use the grid feature to move characters accurately to specific locations on the page. **Managing Data Storage and Files (2.CCT.a.2)** ● Understand how to store and manage files using digital devices. **Algorithms: Connecting a Path (2.DT.a.1, 2.DT.c.1, 2.CCT.c.1)** ● Create and adjust simple algorithms that guide characters based on their starting position, size, and shape. **Careers in CS: Litter Free Communities (2.DT.a.1, 2.DT.c.1, 2.CCT.a.2, 2.CCT.b.3, 2.CCT.c.1)** ● Explain how computer science can help solve community problems by creating a program to collect and sort litter. **Data Patterns and Predictions (2.DT.c.1, 2.CCT.b.1)** ● Identify patterns in data visualizations and use events in a program to show predictions based on the data. **Two-Step Dance & Feedback (2.DC.b.4, 2.DC.c.1, 2.DT.a.1, 2.DT.c.1, 2.CCT.c.1)** ● Revise a program based on peer feedback and provide proper attribution. **Tap-a-Mole Game (2 classes | 2.CCT.c.2)** ● Design an interactive game using event blocks to trigger actions when characters are tapped.

Unit 3: Message Events (4 weeks)

Students will deepen their understanding of program control by using message events to trigger actions, model cycles, and coordinate activity across multiple pages.

Objectives / Topics Covered	● Use message events to control the flow of a program. ● Navigate between pages in a program.
Lessons	**Message Events: Scout Plays in the Forest (2.DT.a.1, 2.DT.c.1, 2.CCT.b.3, 2.CCT.c.1)** ● Use message events to control when and how different parts of a program run. **Programming a Cycle (2.DT.a.1, 2.DT.c.1, 2.CCT.b.3, 2.CCT.c.1)** ● Model a real-world cycle by using message blocks to create repeating sequences in a program. **Pages: Scout's Travels (2.DT.a.1, 2.CCT.c.1)** ● Use message events to move between pages in a multi-scene program. **Exploring Computer Networks (2.CCT.a.2)** ● Understand how networks connect devices and model how messages are sent and received using message blocks.

Unit 4: Loops (5 weeks)

Students will apply loops to identify patterns, build timers, and simplify repeated actions in code. They also strengthen their debugging skills by correcting errors involving loops and message events, and demonstrate creativity by developing and animating an original fiction story.

Objectives / Topics Covered	● Use loops to simplify repeated actions in a program. ● Debug programs by finding and fixing errors.
Lessons	**Loops: Follow the Path (2.DT.a.1, 2.DT.c.1)** ● Identify patterns and use loops to create a program that repeats actions along a path.

	Debugging: Message Events and Loops (2.DT.a.1, 2.CCT.b.3, 2.CCT.c.2) ● Find and fix errors in programs that use both loops and message events. **Making a Timer (2.CCT.b.3, 2.CCT.c.1, 2.CCT.c.2)** ● Use loops, wait blocks, and turn blocks to create and compare timers with different speeds. **Original Story Animations – Fiction Story (2 classes | 2.DT.c.1, 2.CCT.b.3, 2.CCT.c.1)** ● Develop and animate an original fiction story using programming concepts like sequencing and loops.

Unit 5: Culmination Projects (7 weeks)

In this unit, students apply their full range of computer science skills to create interactive games and animations using sequences, events, loops, messages, and pages. They demonstrate their understanding by designing original projects, explaining the purpose of different coding blocks, and showcasing creativity through games.

Objectives / Topics Covered	● Design interactive games that incorporate events, loops, messages, and sequences. ● Demonstrate mastery of computer science concepts through original projects.
Lessons	**Racing Game (2 classes | 2.DT.a.1, 2.DT.c.1, 2.CCT.c.1)** ● Create an interactive racing game using events, loops, and message blocks. **Design an Adventure Game (3 classes | 2.DC.c.1, 2.DT.c.1, 2.CCT.b.3, 2.CCT.c.1, 2.CCT.c.2)** ● Create a multi-page game based on a story. **Collaborating Digitally (2 classes | 2.DT.b.1, 2.CCT.c.1)** ● Collaborate with others digitally to create a classroom program.

Unit 6: Digital Literacy (7 weeks)

Students will learn how to protect personal information with secure login practices, explore searching for information online, and collect and present data.

Objectives / Topics Covered	● Understand the importance of secure login information. ● Conduct research using reliable sources and share findings visually. ● Collect and present data.
Lessons	**Responsible Digital Citizens (2.DC.a.1, 2.DC.a.2, 2.DC.a.3, 2.DC.a.4, 2.DC.b.1, 2.DC.b.3)** ● Understand digital footprints and how to respond to and prevent cyberbullying. **Password Protectors (2.DC.a.2, 2.DC.a.3, 2.DC.a.4)** ● Learn why usernames and passwords are important and practice ways to keep login information safe. **Choice Research (2 classes | 2.DC.b.2, 2.DC.b.4, 2.DT.a.2, 2.DT.c.1, 2.CCT.b.1, 2.CCT.c.1)** ● Research a topic using trusted sources and create a visual program to share findings. **Advanced Data and Programming (3 classes | 2.DT.c.1, 2.CCT.b.1, 2.CCT.b.3, 2.CCT.c.1)** ● Develop an investigative question, collect data, and present data visually.

Optional Preparing for Next Year

In this unit, students will be introduced to CodeHop and programming skills they will learn in 3rd grade.

Objectives / Topics Covered	● Use conditionals to navigate a maze. ● Use variables to keep score in a program. ● Navigate the CodeHop editor.
Lessons	**Coding Card Game: Conditionals** ● Create a sequence of instructions with conditionals to move Scout through a maze. **Coding Card Game: Conditionals 2** ● Use conditionals to help Scout navigate a maze. **Variables: Keeping Score** ● Create a program to simulate keeping score using a variable.

	From CodeHopJr to CodeHop Blocks ● Navigate the CodeHop interface to create a simple program. **From CodeHopJr to CodeHop Blocks: Part 2** ● Use events and loops in CodeHop.

Nova Scotia Information and Communication Technology 2nd 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.
All of these resources and more are found on the **CodeHop Resources Page**.	