Nova Scotia Information and Communication Technology 4th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Nova Scotia Information and Communication Technology 4th 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.

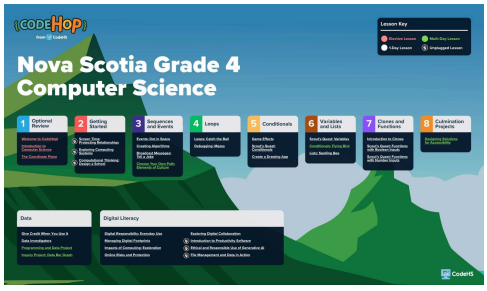
Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging 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**, each 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 CodeHop 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_4/overview



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

Course Breakdown

Optional Review

In this optional unit, students are introduced to key computer science vocabulary, practice logging into and navigating the Playground, and explore basic programming. They also learn to use the coordinate plane to position sprites and create open-ended animations, building spatial reasoning and foundational coding skills.

Objectives / Topics Covered	● Log in and navigate the Playground. ● Understand and use key computer science vocabulary. ● Create a simple program to demonstrate basic programming skills. ● Use the coordinate plane to position sprites.
Lessons	**Welcome to CodeHop!** ● Learn how to log in and explore the Playground to prepare for future lessons. **Introduction to Computer Science** ● Define key computer science vocabulary and create a simple program to apply foundational programming concepts. **The Coordinate Plane** ● Use the coordinate plane to position sprites and create an open-ended animation.

Unit 1: Getting Started (3 weeks)

In this unit, students explore how their online actions can promote positive behavior and develop a code of conduct for responsible internet use. They also apply computational thinking skills to design a school, using strategies like breaking down tasks, identifying patterns, and organizing ideas logically.

Objectives / Topics Covered	● Explore how online actions can promote positivity. ● Apply computational thinking skills. ● Learn about the components and functions of computing systems.
Lessons	**Screen Time: Protecting Relationships (4.DC.a.2)** ● Create a healthy screen-time plan. **Exploring Computing Systems (4.DC.c.3, 4.CT.a.1)** ● Explore the components and functions of computing systems and learn basic troubleshooting techniques. **Computational Thinking: Design a School (4.CT.a.1)** ● Apply computational thinking by breaking down tasks, identifying patterns, and organizing ideas to design a school layout.

Unit 2: Sequences and Events (5 weeks)

In this unit, students deepen their understanding of events by programming with multiple event types and using broadcast messages to coordinate sprite interactions. They also design algorithms and apply coding skills creatively to build interactive projects such as a cultural choose-your-own-path game.

Objectives / Topics Covered	● Create programs using multiple types of event blocks to control when actions happen. ● Develop and compare algorithms to solve problems efficiently. ● Use broadcast messages to coordinate interactions between sprites.
Lessons	**Events: Dot in Space (4.CT.a.2, 4.CT.b.1, 4.CT.c.1, 4.CT.c.3)** ● Create a program using multiple types of event blocks to trigger different actions. **Creating Algorithms (4.CT.a.2, 4.CT.b.3, 4.CT.c.1)** ● Write and compare different algorithms to determine which one best fits the goal of the program.

	Broadcast Messages: Tell a Joke (4.CT.a.2, 4.CT.b.1, 4.CT.c.1, 4.CT.c.3) ● Use broadcast messages to program two sprites that take turns telling a knock-knock joke. **Choose Your Own Path: Elements of Culture (2 classes | 4.DC.b.3, 4.DT.b.1, 4.CT.c.2)** ● Create a choose-your-own-path game that explores elements of culture through branching storylines and interactive programming.

Unit 3: Loops (2 weeks)

In this unit, students learn to use different types of loops to create a simple interactive game and understand how repetition can simplify code. They also practice debugging by breaking down and analyzing a program to identify and fix errors, strengthening their problem-solving skills.

Objectives / Topics Covered	● Use different types of loops to simplify algorithms. ● Break down and analyze code to identify and fix errors.
Lessons	**Loops: Catch the Ball (4.CT.a.2, 4.CT.c.1, 4.CT.c.3)** ● Use two types of loops to create a simple interactive game where players catch a moving ball. **Debugging: Mazes (4.CT.a.2, 4.CT.c.1, 4.CT.c.3)** ● Break down and analyze a program to find and fix errors, focusing on loops and logical flow.

Unit 4: Conditionals (3 weeks)

Students explore conditionals to control the behavior of sprites in their programs. Students will use these skills to add effects to their programs and to develop interactive games.

Objectives / Topics Covered	● Use if/then conditionals to control decision-making in programs. ● Use keyboard and mouse inputs.
Lessons	**Game Effects (4.DC.c.2, 4.DT.c.3, 4.CT.a.2, 4.CT.c.1, 4.CT.c.2)** ● Modify a game's effects and make updates based on feedback. **Scout's Quest: Conditionals (4.CT.a.2, 4.CT.c.1)** ● Use if/then conditionals to build a program that makes decisions based on input or events. **Create a Drawing App (4.CT.a.2, 4.CT.c.1)** ● Program keyboard and mouse inputs using loops and conditionals to create a drawing app.

Unit 5: Variables and Lists (4 weeks)

Students explore variables to store and change values while a program is running. Students will also explore lists to store more data in a program.

Objectives / Topics Covered	● Learn how variables store information. ● Create and change the value of variables.
Lessons	**Scout's Quest: Variables (4.CT.a.2, 4.CT.c.1, 4.CT.c.3)** ● Create and use variables to track information such as points in a game. **Conditionals: Flying Bird (2 classes | 4.CT.a.2, 4.CT.c.1)** ● Use conditionals to control a flying bird game. **Lists: Spelling Bee (4.CT.a.2, 4.CT.c.1)** ● Use lists to hold data in a spelling bee game.

Unit 6: Clones and Functions (3 weeks)

In this unit, students explore how to use clones and functions to create more efficient and dynamic programs. They build animations, design custom functions with inputs—including booleans and numbers—and apply these concepts creatively in projects like drawing with functions and building interactive features based on user input.

Objectives / Topics Covered	● Explore how clones affect program behavior. ● Use functions with number and boolean inputs to trigger actions based on conditions.
Lessons	**Introduction to Clones (4.CT.a.2, 4.CT.c.1)** ● Create an animation using clones and explore how clone behavior affects the program's outcome. **Scout's Quest: Functions with Boolean Inputs (4.CT.a.2, 4.CT.c.1, 4.CT.c.3)** ● Build a function that uses a boolean input to perform different actions, such as checking if a password is correct. **Scout's Quest: Functions with Number Inputs (4.CT.a.2, 4.CT.b.1, 4.CT.c.1, 4.CT.c.3)** ● Use number inputs in a function to create a drawing that changes based on input values.

Unit 7: Culmination Project (2 weeks)

Students apply their programming knowledge to modify a game for user accessibility, using events, conditionals, variables, comparison operators, and broadcast messages to bring their project to life. This final project reinforces key concepts and allows for creativity and problem-solving in a self-directed build.

Objectives / Topics Covered	● Apply key programming concepts including conditionals, variables, and events. ● Demonstrate mastery of coding skills through creative problem-solving and game design.
Lessons	**Designing Solutions for Accessibility (2 classes | 4.DC.c.2, 4.DC.d.1, 4.DT.c.2, 4.CT.c.4)** ● Explore how to apply design thinking to create digital tools and experiences that are accessible to everyone.

Unit 8: Data (6 weeks)

Students will explore how to search for, analyze, collect, and present data to answer questions and solve problems.

Objectives / Topics Covered	● Search for information responsibly online. ● Evaluate and analyze data. ● Collect data and display results visually.
Lessons	**Give Credit When You Use It (4.DC.b.3, 4.DT.a.1, 4.DT.a.2, 4.DT.a.3, 4.DT.c.1)** ● Search for information online and provide proper attribution. **Data Investigators (4.DT.a.1, 4.CT.b.2)** ● Evaluate data for reliability and analyse it to draw conclusions. **Programming and Data Project (2 classes | 4.DC.c.2, 4.CT.b.2, 4.CT.c.4)** ● Collect data to draw conclusions and present them in a program. **Inquiry Project: Data Bar Graph (2 classes | 4.DT.a.3, 4.DT.b.1, 4.DT.c.3, 4.CT.a.2, 4.CT.b.2)** ● Follow the inquiry process to collect data and modify a program to display the results using a bar graph.

Unit 9: Digital Literacy (8 weeks)

In this unit, students develop digital literacy by exploring safe online behaviors. They investigate technology's impact on culture and begin to explore concepts such as digital ownership, accessibility, and cybersecurity.

Objectives	● Practice safe online behavior.

/ Topics Covered	• Select different online tools to complete tasks.
Lessons	**Digital Responsibility: Everyday Use (4.DC.a.1, 4.DC.a.3)** • Identify online dangers and demonstrate how to make responsible choices online. **Managing Digital Footprints (4.DC.a.4, 4.DC.b.1)** • Explain how online actions create permanent digital footprints. **Impacts of Computing: Exploration (4.DC.c.3, 4.DC.d.2)** • Explore how computing affects individuals, society, and the world around us. **Online Risks and Protection (4.DC.a.4)** • Practice secure habits and understand tools that protect your information. **Exploring Digital Collaboration (4.DT.b.2)** • Collaborate with others digitally to complete a program that solves a community problem. **Introduction to Productivity Software (4.DC.c.3, 4.DT.c.1)** • Compare and select appropriate software applications to complete different tasks. **Ethical and Responsible Use of Generative AI (4.DC.b.2, 4.DC.d.2)** • Complete a class code of conduct for using AI. **File Management and Data in Action (4.DC.c.1)** • Organize, save, and manage digital files effectively. • *Note: the extension must be completed to meet all standards in this lesson.*

Nova Scotia Information and Communication Technology 4th 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.
All of these resources and more are found on the **CodeHop Resources Page**.	