# Nova Scotia Information and Communication Technology 3rd Grade Course Syllabus

### One Year for Elementary School, 36 Hours

## Course Overview and Goals

The **Nova Scotia Information and Communication Technology 3rd Grade Course** introduces students to foundational programming concepts through **Scratch**, 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.

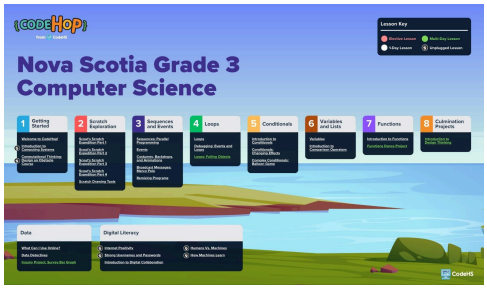
**Learning Environment:** This course is designed to be teacher-led, with ready-to-use lesson plans that follow a structured format: **Introduction, Guided Practice, Independent Practice, Extension, and Reflection**. Lessons are built with spiral review to reinforce key concepts and culminate in engaging 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 **Scratch** 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_3/overview




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

## Course Breakdown

**Unit 1: Getting Started (2 weeks)**

In this introductory unit, students are introduced to the basics of computing by identifying key parts of a computing system and learning how to troubleshoot simple issues. They also begin developing foundational computational thinking skills by designing and sequencing steps in an obstacle course challenge.

| | |
|---|---|
| Objectives / Topics Covered | ● Log in and navigate the Playground.<br>● Identify parts of a computing system.<br>● Practice basic troubleshooting strategies for common computer problems.<br>● Apply computational thinking skills. |
| Lessons | **Welcome to CodeHop!**<br>● Learn how to log in and navigate the Playground to get comfortable using the platform.<br>**Introduction to Computing Systems**<br>● Identify the main parts of a computing system—hardware, software, input, and output—and solve basic problems when something doesn't work.<br>**Computational Thinking: Design an Obstacle Course**<br>● Use computational thinking skills like sequencing and problem-solving to plan and design an obstacle course. |

**Unit 2: Scratch Exploration (4 weeks)**

Students will be introduced to the Scratch editor in this story-driven introductory unit. Throughout this unit students will practice creating animations using loops, events, and sequences.

| | |
|---|---|
| Objectives / Topics Covered | ● Navigate the Scratch editor.<br>● Use events, loops, and motion blocks.<br>● Create an animated story. |
| Lessons | **Scout's Scratch Expedition Part 1**<br>● Use basic Scratch commands to program a sprite.<br>**Scout's Scratch Expedition Part 2**<br>● Create a sequence to animate a story.<br>**Scout's Scratch Expedition Part 3**<br>● Use loops, events, looks, and motion blocks to create an animated story.<br>**Scout's Scratch Expedition Part 4**<br>● Create an animated story in Scratch. |

**Unit 3: Sequences and Events (5 weeks)**

Students build foundational programming skills by creating sequences and using events to control when actions happen. They also explore parallel programming and broadcast messages to coordinate interactions between sprites.

| | |
|---|---|
| Objectives / Topics Covered | ● Create programs using sequences.<br>● Use event blocks to trigger actions and coordinate program flow.<br>● Program simple animations and interactions. |
| Lessons | **Sequences: Parallel Programming**<br>● Create a program with multiple sequences running at the same time to control different sprite actions.<br>**Events**<br>● Use event blocks to trigger actions in a Scratch program and control when things happen.<br>**Costumes, Backdrops, and Animations**<br>● Animate sprites and backdrops in a program. |

| | |
|---|---|
| | **Broadcast Messages: Marco Polo**<br>● Use broadcast messages to trigger action between sprites.<br>**Remixing Programs**<br>● Create or remix digital projects using appropriate content while giving credit to original creators. |

**Unit 4: Loops (4 weeks)**

Students learn how loops repeat instructions and use them to create more efficient programs in Scratch. They also build debugging skills by analyzing and fixing errors in programs, and compare different types of loops to understand their effects and uses in animations.

| | |
|---|---|
| Objectives / Topics Covered | ● Use loops to simplify code.<br>● Identify and fix issues related to loops and events.<br>● Develop efficient, repeatable patterns in animations and interactive projects. |
| Lessons | **Loops**<br>● Learn that loops repeat one or more instructions and use them in Scratch to simplify and improve programs.<br>**Debugging: Events and Loops**<br>● Decompose a program to debug and make the program run as intended.<br>**Loops: Falling Objects (2 part lesson)**<br>● Create a program using different types of loops and compare how each affects how the program runs. |

**Unit 5: Conditionals (2 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 and if/else blocks in Scratch programs. |
| Lessons | **Introduction to Conditionals**<br>● Learn how to use if/then blocks to make decisions in a program based on specific conditions.<br>**Conditionals: Changing Effects**<br>● Create a program using conditionals. |

**Unit 6: Variables and Lists (2 weeks)**

Students explore variables to store and change values while a program is running. Students will also explore comparison operators to develop more complex decision making within a program.

| | |
|---|---|
| Objectives / Topics Covered | ● Learn how variables store information.<br>● Create and change the value of variables.<br>● Use comparison operators to create more complex decision-making logic. |
| Lessons | **Variables**<br>● Create and change the value of a variable.<br>**Introduction to Comparison Operators**<br>● Use comparison operators with numbers and variables to create more complex if/else conditions. |

**Unit 7: Functions (4 weeks)**

Students will learn how to use functions to make their programs more organized and efficient. Then, students will apply this knowledge to create a dance animation.

| | |
|---|---|
| Objectives / Topics Covered | ● Use functions to organize code into reusable parts. |
| Lessons | **Introduction to Functions**<br>● Create and use functions in a Scratch program to organize code and make actions easier to repeat and manage.<br>**Functions Dance Project (3 part lesson)**<br>● Create and use functions to call dance moves in a sequence. |

**Unit 8: 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 | ● Design, create, and personalize a project that demonstrates mastery of core coding skills. |
| Lessons | **Introduction to Design Thinking (2 part lesson)**<br>● Explore ways to make digital tools more accessible using the design process. |

**Unit 9: Data (4 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.<br>● Evaluate and analyze data.<br>● Collect data and display results visually. |
| Lessons | **What Can I Use Online?**<br>● Research information to answer questions online and give credit to original sources.<br>**Data Detectives**<br>● Evaluate data for reliability and analyse it to draw conclusions.<br>**Inquiry Project: Survey Bar Graph (2 part lesson)**<br>● Follow the inquiry process to collect data and modify a program to display the results using a bar graph. |

**Unit 10: Digital Literacy (7 weeks)**

Students explore how their actions can affect people and learn strategies for staying safe online, including creating strong usernames and passwords and recognizing common cybersecurity threats.

| | |
|---|---|
| Objectives / Topics Covered | ● Understand how to balance screen time in a healthy way.<br>● Learn strategies for staying safe online.<br>● Collaborate effectively online.<br>● Understand approaches to machine learning. |
| Lessons | **My Screen Balance Plan**<br>● Develop healthy technology routines. |

| | |
|---|---|
| | **Introduction to Digital Etiquette and Communication**<br>● Demonstrate proper digital etiquette when communicating in an online community.<br>**Cybersecurity Introduction**<br>● Identify common cyber threats and learn tips for staying safe online.<br>**Digital Footprint Basics**<br>● Describe safe and unsafe online behaviors and explain how those choices affect a digital footprint.<br>**Introduction to Digital Collaboration**<br>● Collaborate with others digitally to complete a program that promotes a community event.<br>**Humans Vs. Machines**<br>● Compare and contrast human and computer performance on similar tasks.<br>**How Machines Learn**<br>● Explain the different machine learning approaches. |

## Nova Scotia Information and Communication Technology 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 Scratch Templates | Empower your students to explore and express their knowledge creatively with our versatile Scratch 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**. | |