



Nebraska Technology Kindergarten Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Nebraska Technology Kindergarten 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 course includes a total of 36 **contact hours**, with each lesson approximately 30 minutes long. This provides a full school year of material if teaching one lesson per week. Digital literacy and productivity software lessons are also available to complement the programming curriculum with non-programming computer and technology skills. 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/NE_K/overview



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

Course Breakdown

Optional Unplugged Exploration

In this unplugged unit, students develop foundational programming skills such as sequencing, pattern recognition, and algorithmic thinking through hands-on activities that encourage collaboration, problem-solving, and computational reasoning.

Objectives / Topics Covered	• Understand and create simple sequences through hands-on activities. • Practice giving and following directions. • Recognize patterns and develop problem-solving skills.
Lessons	Sequences (Unplugged) • Create a step-by-step sequence of dance moves to understand how order matters in instructions. Coding Card Game: Sequences • Work together to give clear directions that move a character through a maze, practicing how to build a sequence of instructions. Coding Card Game: Sequences 2 • Build on the previous activity by creating new sequences and solving different mazes as a team. Acting with Events • Act out simple sequences where something happens to cause an action, introducing the idea of events in a fun and physical way.

Unit 1: Getting Started (5 weeks)

In this introductory unit, students will begin developing basic computer skills and explore foundational concepts in computer science. Through hands-on practice and familiar routines, they'll learn how computers work, how to use a mouse and keyboard, and how to think like a computer scientist by recognizing patterns and sequencing steps.

Objectives / Topics Covered	• Learn how to log in and navigate the Playground. • Understand what a computer is and how its parts work together. • Practice using a mouse and keyboard. • Apply computational thinking to everyday routines.
Lessons	Welcome to CodeHop! (K.BT.8, K.BT.11, K.DM.1) • Introductory lesson to help students log in and explore the Playground; perfect as a warm-up or standalone activity. Mouse Practice (K.BT.9) • Practice mouse skills like clicking and dragging through fun and interactive games. Keyboard Introduction • Explore the keyboard by identifying and using letters, numbers, and simple function keys. Computer Basics: Introduction (K.BT.7, K.BT.9, K.BT.12, K.BT.14, K.BT.16) • Learn what a computer is, how it's used, and how input, output, hardware, and software work together. Introduction to Responsible Technology Use (K.DC.1, K.DC.6, K.DC.7) • Identify ways to use technology safely and responsibly. Computational Thinking: Morning Routines (K.CS.1) • Break down a morning routine into steps and recognize patterns using computational thinking strategies.

Unit 2: Programming Exploration (6 weeks)

In this story-based unit, students will explore the basics of block-based programming, learning how to animate characters, build simple stories, and begin thinking like programmers through experimentation and play.

Objectives / Topics Covered	• Learn to navigate the CodeHopJr interface. • Add, delete, and modify characters and backgrounds. • Create sequences. • Introduce event blocks to trigger movement and interactions.
-----------------------------------	--

Lessons	Scout Adventures 1: Introducing Scout - Explore the CodeHopJr interface and add characters to a project. Scout Adventures 2: Scout Starts Exploring - Add backgrounds and a new page to a project. Scout Adventures 3: Scout Meets a Friend - Delete and modify characters. Scout Adventures 4: Scout Explores the Forest (K.CS.1) - Use motion blocks to move characters around the page. Scout Adventures 5: Scout and Bluebird Help (K.CS.1) - Build a sequence of motion blocks to move characters and collect objects. Scout Adventures 6: Scout Celebrates with Friends (K.CS.1) - Create a celebration scene by adding characters, pages, backgrounds, and sequences of motion blocks with events.
---------	---

Unit 3: Sequences and Events (5 weeks)

In this unit, students will dive deeper into computer science by learning how to create precise sequences of instructions and use events to control when actions happen in a program.

Objectives / Topics Covered	- Create sequences using motion and event blocks. - Use events to trigger animations and interactions. - Use painting tools to create unique characters and scenes.
Lessons	Drawing Tools: Fairy Tale Painting (K.BT.9) - Use painting tools to create a fairy-tale scene. Introduction to Events (K.CS.1) - Create a program using different types of events. Sequences: Snowball Fight (K.CS.1) - Create a program using multiple sequences. What Can Data Tell Us? (K.R.4) - Collect, organize, and explore data about how students get to school, using charts or visuals to understand patterns. Careers in CS: Coding for Fashion-Retail (K.DC.6, K.R.2, K.CS.1) - Explain how coding helps create and improve fashion designs by programming a fashion character.

Unit 4: Pages (3 weeks)

In this unit, students will use Pages to organize ideas across multiple scenes, learning how to structure digital stories, plan projects, and develop computational thinking skills.

Objectives / Topics Covered	- Create multi-page programs. - Use the “go to page” block to switch between pages.
Lessons	Introduction to Pages (K.BT.10) - Create a program with multiple pages. Pages: Dragon Story (K.BT.10) - Create a multi-page story featuring a dragon, with different actions and scenes on each page. Using the Go To Page Block (K.CS.1) - Build a program that switches between pages using the “go to page” block for interactive navigation.

Unit 5: Block Exploration (3 weeks)

In this unit, students will investigate how different types of programming blocks work together, deepening their understanding of commands, control, motion, and how to combine them to create engaging projects.

Objectives / Topics Covered	• Use speed, sound, and say blocks. • Identify and fix errors in code.
Lessons	Introduction to Speed Blocks (K.CS.1) • Create a program that uses different speed blocks to animate characters. Animal Sounds (K.CS.1) • Use "say" or "sound" blocks to program a character to speak. Debugging: Events and Motion (K.CS.1) • Find and fix bugs in a program that uses event and motion blocks.

Unit 6: Loops (3 weeks)

In this unit, students will explore the concept of loops, learning how to use repetition in programs to make their code more efficient and solve problems more creatively.

Objectives / Topics Covered	• Use loops to simplify sequences. • Recognize patterns that can be repeated in code.
Lessons	Loops • Create a program using loops and explain how loops are used to repeat code. Loops: Frog and Rabbit (K.CS.1) • Use loops to animate a frog and rabbit repeating actions in a fun, interactive program. Forever Loops: Fireworks (K.CS.1) • Create sequences that repeat continuously while the program is running, using forever loops to animate ongoing actions.

Unit 7: Culmination Project (2 weeks)

In this unit, students will apply what they've learned by designing and creating their own animation showcasing their programming skills, creativity, and understanding of computer science concepts.

Objectives / Topics Covered	• Use sequences, events, and loops in an original project.
Lessons	Wildlife Scene Project (2 classes K.CS.1) • Design a wildlife animation using events, sequences, and loops to bring the scene to life.

Unit 8: Digital Literacy (7 weeks)

In this unit, students will learn the basics of digital citizenship, including how to navigate technology safely, make responsible choices online, and understand the roles computers play in their lives and communities.

Objectives / Topics Covered	• Use technology safely and responsibly. • Explore the role of computers and the internet in everyday life. • Collect, organize, and analyze data through surveys and online research.
Lessons	Be Kind Online (K.DC.5) • Understand how to communicate positively online. Keeping Information Safe (K.DC.2, K.DC.6, K.DC.7) • Identify the difference between personal and private information and learn how to keep information safe.

	Introduction to Research (2 classes K.BT.10, K.R.1, K.R.3, K.R.5) • Use simple research tools to find information and create a project that shares findings visually. Introduction to Data Storage and Files (K.BT.5, K.BT.6) • Understand how to store and manage files using digital devices. Using Networks to Connect (K.BT.15, K.DC.6) • Learn how people and devices connect and share information using networks. What Can AI Do? (K.R.6) • Identify tools that use AI and compare tasks that are better suited for humans vs AI.
--	--

Unit 9: Productivity Software (2 weeks)

In this unit, students will be introduced to the different types of productivity software and will practice using word processing skills.

Objectives / Topics Covered	• Understand the purposes of different types of productivity software. • Practice using word processing software.
Lessons	Types of Software (K.BT.10, K.BT.12, K.R.1) • Select appropriate software applications for different tasks. Introduction to Word Processing (K.BT.12, K.P.1) • Create a document about favorite seasons.

Nebraska Technology Kindergarten 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 .	