Alabama Digital Literacy and Computer Science: Kindergarten Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The **Alabama Digital Literacy and Computer Science: Kindergarten Course** introduces students to foundational programming concepts through a block-based programming language. Students explore digital literacy and computer science skills while learning to create interactive projects, animations, and games. This course emphasizes creativity, collaboration, and real-world connections, providing students with a strong foundation in computer science concepts and digital literacy.

Learning Environment: This course is designed to be teacher-led, with ready-to-use lesson plans. Each programming lesson follows a structured format: **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. Many digital literacy lessons contain unplugged activities, requiring printed handouts and class activities to support hands-on learning.

The course includes a total of **36 lessons,** each approximately 30-45 minutes long. 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.

Standards Alignment Note: Lessons that list "*Standards Met*" (below) are required to fully meet state computer science standards. Lessons without a standards tag support spiral review, practice, or enrichment.

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/AL_K/overview

A clickable PDF can be found at https://codehs.com/AL-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! (15 minute lesson)** ● Introductory lesson to help students log in and explore the Playground; perfect as a warm-up or standalone activity. **Mouse Practice** ● Practice mouse skills like clicking and dragging through fun and interactive games. ● *Standard Met: DLCS25.K.13* **Keyboard Introduction** ● Explore the keyboard by identifying and using letters, numbers, and simple function keys. ● *Standard Met: DLCS25.K.13* **Computer Basics: Introduction** ● Learn what a computer is, how it's used, and how input, output, hardware, and software work together. ● *Standards Met: DLCS25.K.9, DLCS25.K.13* **Introduction to Responsible Technology Use** ● Identify ways to use technology safely and responsibly. ● *Standard Met: DLCS25.K.8* **Computational Thinking: Morning Routines** ● Break down a morning routine into steps and recognize patterns using computational thinking strategies. ● *Standards Met: DLCS25.K.1, DLCS25.K.2, DLCS25.K.3*

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** ● Use motion blocks to move characters around the page. **Scout Adventures 5: Scout and Bluebird Help** ● Build a sequence of motion blocks to move characters and collect objects. **Scout Adventures 6: Scout Celebrates with Friends** ● Create a celebration scene by adding characters, pages, backgrounds, and sequences of motion blocks with events.

Unit 3: Sequences and Events (8 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** ● Use painting tools to create a fairy-tale scene. **Introduction to Events** ● Create a program using different types of events. **Sequences: Snowball Fight** ● Create a program using multiple sequences. ● *Standards Met: DLCS25.K.1, DLCS25.K.3* **What Can Data Tell Us?** ● Collect, organize, and explore data about how students get to school, using charts or visuals to understand patterns. ● *Standard Met: DLCS25.K.4* **Events: Submarine Sequences** ● Use event and motion blocks to program an underwater scene. ● *Standards Met: DLCS25.K.1, DLCS25.K.3* **Introduction to Show and Hide Blocks (2 part lesson)** ● Make characters appear and disappear using “show” and “hide” blocks. **Introduction to Grow and Shrink Blocks** ● Change the size of characters using “grow” and “shrink” blocks.

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** • Create a program with multiple pages. **Pages: Dragon Story** • Create a multi-page story featuring a dragon, with different actions and scenes on each page. **Using the Go To Page Block** • 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** • Create a program that uses different speed blocks to animate characters. **Animal Sounds** • Use "say" or "sound" blocks to program a character to speak. **Debugging: Events and Motion** • Find and fix bugs in a program that uses event and motion blocks.

Unit 6: Loops (2 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. **Forever Loops: Fireworks** • 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 part lesson)** • 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 and make responsible choices online. Students will also explore different types of technology including productivity software, networks, and artificial intelligence.

Objectives / Topics Covered	• Use technology safely and responsibly. • Search for information online. • Explore different types of software. • Discuss how networks help people connect. • Define artificial intelligence.
Lessons	**Keeping Information Safe** • Identify the difference between personal and private information and learn how to keep information safe. • *Standards Met: DLCS25.K.7, DLCS25.K.8* **Introduction to Research (2 part lesson)** • Use simple research tools to find information and create a project that shares findings visually. • *Standard Met: DLCS25.K.11* **Types of Software** • Select appropriate software applications to complete different tasks. • *Standard Met: DLCS25.K.14* **Using Networks to Connect** • Describe how people and devices connect and share information using a network. • *Standards Met: DLCS25.K.6, DLCS25.K.12* **What is AI?** • Identify examples of AI and explain how smart machines are different from other tools. • *Standards Met: DLCS25.K.10, DLCS25.K.15* **Sorting with Decision Trees** • Explain how AI uses data to learn and make decisions. • *Standards Met: DLCS25.K.4, DLCS25.K.5, DLCS25.K.10*

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.
These resources and more are found on the **CodeHop Resources Page**.	