# Alabama Digital Literacy and Computer Science: 3rd Grade Course Syllabus

### One Year for Elementary School, 36 Hours

## Course Overview and Goals

The **Alabama Digital Literacy and Computer Science: 3rd Grade 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.

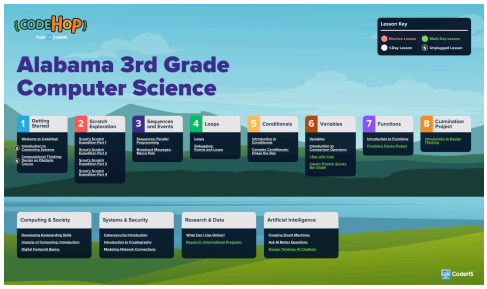
The course includes a total of **36 lessons,** each approximately 45 minutes long. The concepts taught in this course spiral across grade levels, ensuring that students can revisit and build upon their understanding year after year, even if all lessons are not completed within a single year.

**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_3/overview




A clickable PDF can be found at https://codehs.com/AL-K-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>● *Standard Met: DLCS25.3.10*<br>**Computational Thinking: Design an Obstacle Course**<br>● Use computational thinking skills like sequencing and problem-solving to plan and design an obstacle course.<br>● *Standard Met: DLCS25.3.1* |

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

Students explore the basics of CodeHop by using commands to move and animate sprites, building familiarity with sequencing, motion, and dialogue. As they progress through a story-driven unit, they apply programming concepts such as events and loops to create interactive and animated stories.

| | |
|---|---|
| Objectives / Topics Covered | ● Use basic commands to program sprites.<br>● Explore the CodeHop interface. |
| Lessons | **Scout's Programming Expedition Part 1**<br>● Use basic commands to program a sprite to move and talk while beginning a story-driven coding adventure.<br>**Scout's Programming Expedition Part 2**<br>● Add new sprites and create a sequence of actions to animate a simple story.<br>**Scout's Programming Expedition Part 3**<br>● Create an animated story using loops, events, looks, and motion blocks to bring characters to life.<br>**Scout's Programming Expedition Part 4**<br>● Continue building an animated story by using events, looks, and motion blocks to enhance interactivity and animation. |

**Unit 3: Sequences and Events (2 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>● Apply broadcast messages to enable communication between multiple sprites. |

| | |
|---|---|
| Lessons | **Sequences: Parallel Programming**<br>● Create a program with multiple sequences running at the same time to control different sprite actions.<br>● *Standard Met: DLCS25.3.3*<br>**Broadcast Messages: Marco Polo**<br>● Use broadcast messages to make sprites communicate and respond to each other's actions.<br>● *Standards Met: DLCS25.3.3, DLCS25.3.5* |

**Unit 4: Loops (2 weeks)**

Students learn how loops repeat instructions and use them to create more efficient programs. They also build debugging skills by analyzing and fixing errors in programs.

| | |
|---|---|
| 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 to simplify and improve programs.<br>**Debugging: Events and Loops**<br>● Decompose a program to debug and make the program run as intended. |

**Unit 5: Conditionals (2 weeks)**

Students explore conditional logic to make programs respond dynamically to different conditions and inputs.

| | |
|---|---|
| Objectives / Topics Covered | ● Use if/then and if/else blocks.<br>● Use conditionals to build interactive and reactive programs. |
| Lessons | **Introduction to Conditionals**<br>● Learn how to use if/then blocks to make decisions in a program based on specific conditions.<br>**Complex Conditionals: Chase the Star**<br>● Use if/then/else conditionals in a program.<br>● *Standard Met: DLCS25.3.4* |

**Unit 6: Variables (6 weeks)**

In this unit, students explore variables and comparison operators to manage and organize data in programs.

| | |
|---|---|
| Objectives / Topics Covered | ● Learn how variables store information.<br>● Use comparison operators to create more complex decision-making logic. |
| Lessons | **Variables**<br>● Understand what variables are and how to create and update them to store changing information in a program.<br>**Introduction to Comparison Operators**<br>● Use comparison operators with numbers and variables to create more complex if/else conditions.<br>**I-Spy with Lists (2 part lesson)**<br>● Use variables and lists to create an I-Spy game.<br>**Inquiry Project: Survey Bar Graph (2 part lesson)**<br>● Follow the inquiry process and modify a program to display the results of an investigation.<br>● *Standard Met: DLCS25.3.6* |

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

Students will explore functions in this unit, allowing them to refine and organize more complex programs.

| | |
|---|---|
| Objectives / Topics Covered | ● Use functions to organize code into reusable parts. |
| Lessons | **Introduction to Functions**<br>● Create and use functions to organize code and make actions easier to repeat and manage.<br>**Functions Dance Project (3 part lesson)**<br>● Use functions to call dance moves in a sequence that aligns with music.<br>● *Standard Met: DLCS25.3.3* |

**Unit 8: Culmination Project (2 weeks)**

Students apply their programming knowledge to make a tool more accessible to users, with 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.<br>● *Standards Met: DLCS25.3.3, DLCS25.3.5, DLCS25.3.13* |

**Unit 9: Computing & Society (2 weeks)**

In this unit, students explore how different computing technologies impact the world around them and gain an understanding of how to use digital tools responsibly.

| | |
|---|---|
| Objectives / Topics Covered | ● Understand how technology impacts society.<br>● Learn strategies for staying safe online. |
| Lessons | **Developing Keyboarding Skills (Ongoing typing program)**<br>● Use proper finger placement to practice typing on a keyboard.<br>● *Standard Met: DLCS25.3.19*<br>**Impacts of Computing: Introduction**<br>● Explore how computing affects people and society.<br>● *Standard Met: DLCS25.3.11*<br>**Digital Footprint Basics**<br>● Explain how choices made online can affect a digital footprint.<br>● *Standards Met: DLCS25.3.15, DLCS25.3.16* |

**Unit 10: Systems & Security (3 weeks)**

In this unit, students will learn how to stay safe online and explore how networks connect devices.

| | |
|---|---|
| Objectives / Topics Covered | ● Recognize common digital threats.<br>● Understand how messages move between devices using a network. |

| | |
|---|---|
| Lessons | **Cybersecurity Introduction**<br>● Identify common cyber threats and learn tips for staying safe online.<br>● *Standards Met: DLCS25.3.8, DLCS25.3.9*<br>**Introduction to Cryptography**<br>● Learn basic cryptography to solve multiple ciphers.<br>● *Standard Met: DLCS25.3.2*<br>**Modeling Network Connections**<br>● Create a program that shows how data travels from one device to another.<br>● *Standard Met: DLCS25.3.7* |

**Unit 11: Research & Data (3 weeks)**

In this unit, students will learn how to search for trustworthy information online and communicate what they learn through interactive programs.

| | |
|---|---|
| Objectives / Topics Covered | ● Follow laws when researching online.<br>● Communicate research findings. |
| Lessons | **What Can I Use Online?**<br>● Research information to answer questions online and give credit to original sources.<br>● *Standard Met: DLCS25.3.14*<br>**Research: Informational Programs (2 part lesson)**<br>● Examine and communicate information from multiple online sources.<br>● *Standard Met: DLCS25.3.6* |

**Unit 12: Artificial Intelligence (4 weeks)**

In this unit, students will learn how to use AI tools responsibly and effectively.

| | |
|---|---|
| Objectives / Topics Covered | ● Describe how AI representations support reasoning.<br>● Describe how emerging technologies can be used in daily life.<br>● Write AI prompts. |
| Lessons | **Creating Smart Machines**<br>● Describe how emerging technologies can be used and why people design new technology.<br>● *Standard Met: DLCS25.3.12*<br>**Ask AI Better Questions**<br>● Write clear prompts for AI and decide if the responses are reasonable.<br>● *Standard Met: DLCS25.3.12, DLCS25.3.18*<br>**Design Thinking: AI Chatbots (2 part lesson)**<br>● Train a model AI chatbot program.<br>● Standard Met: DLCS25.3.17 |

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