Maryland Computer Science
1st Grade Course Syllabus
One Year for Elementary School, 36 Hours

Course Overview and Goals

The Maryland Computer Science 1st 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 teacher-led and includes ready-to-use lessons following a consistent structure: 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.

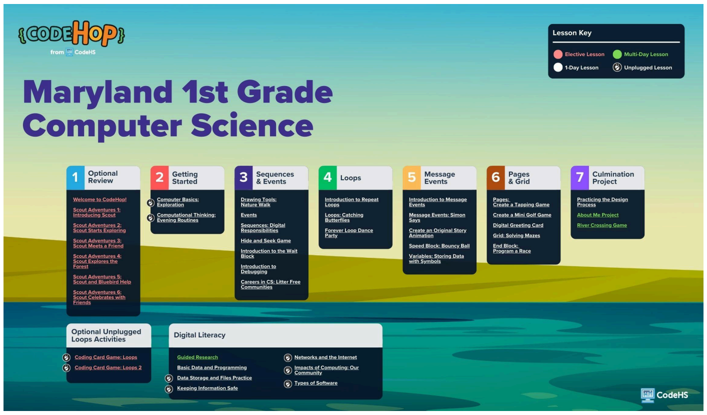
The course includes 36 lessons, each approximately 45 minutes long, providing a full year of instruction when taught once per week. While the course allows for instructional flexibility, some lessons are required to fully meet state computer science standards and are clearly identified within the syllabus. All Digital Literacy lessons are required to ensure full standards alignment, as they address essential non-programming computer science concepts. Required lessons are labeled with the specific standards they address to support planning and compliance.

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 the CodeHop platform. 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/27665/overview



Maryland 1st Grade Computer Science

Lesson Key
- 🟥 Elective Lesson
- 🟩 Multi-Day Lesson
- ○ 1-Day Lesson
- ✋ Unplugged Lesson

1 Optional Review
- Welcome to CodeHop!
- Scout Adventures 1: Introducing Scout
- Scout Adventures 2: Scout Starts Exploring
- Scout Adventures 3: Scout Meets a Friend
- Scout Adventures 4: Scout Explores the Forest
- Scout Adventures 5: Scout and Bluebird Help
- Scout Adventures 6: Scout Celebrates with Friends

2 Getting Started
- ✋ Computer Basics: Exploration
- ✋ Computational Thinking: Evening Routines

3 Sequences & Events
- Drawing Tools: Nature Walk
- Events
- Sequences: Digital Responsibilities
- Hide and Seek Game
- Introduction to the Wait Block
- Introduction to Debugging
- Careers in CS: Litter Free Communities

4 Loops
- Introduction to Repeat Loops
- Loops: Catching Butterflies
- Forever Loop Dance Party

5 Message Events
- Introduction to Message Events
- Message Events: Simon Says
- Create an Original Story Animation
- Speed Block: Bouncy Ball
- Variables: Storing Data with Symbols

6 Pages & Grid
- Pages: Create a Tapping Game
- Create a Mini Golf Game
- Digital Greeting Card
- Grid: Solving Mazes
- End Block: Program a Race

7 Culmination Project
- Practicing the Design Process
- About Me Project
- River Crossing Game

Optional Unplugged Loops Activities
- ✋ Coding Card Game: Loops
- ✋ Coding Card Game: Loops 2

Digital Literacy
- Guided Research
- Basic Data and Programming
- ✋ Data Storage and Files Practice
- ✋ Keeping Information Safe
- ✋ Networks and the Internet
- ✋ Impacts of Computing: Our Community
- ✋ Types of Software

A clickable PDF can be found at https://codehs.com/MD-CSRoadmaps

Course Breakdown

Optional Review

This optional module supports students in revisiting key skills such as logging in, navigating the coding environment, and reviewing basic programming concepts introduced in previous lessons.

Objectives / Topics Covered	● Refresh familiarity with the CodeHop Playground and login process. ● Revisit character, page, and background creation. ● Reinforce sequences and motion block usage in a narrative structure.
Lessons	**Welcome to CodeHop! (15 minute lesson)** ● Learn how to log in and navigate the Playground. Can be used as a warm-up or review. **Scout Adventures 1-6** ● Follow Scout the Squirrel through a six-part interactive story that helps students build foundational skills.

Unit 1: Getting Started (2 lessons)

Students are introduced to basic computer concepts and computational thinking practices. This unit establishes a foundation in technology use, computer literacy, and problem-solving skills that students will carry into later lessons.

Objectives / Topics Covered	● Identify the parts of a computer and explain how we use them. ● Apply computational thinking skills to everyday routines.
Lessons	**Computer Basics: Exploration** ● Learn how computers work and understand hardware, software, input, and output. ● *Standards Met: 1.CS.HS.01, 1.CS.T.01* **Computational Thinking: Evening Routines** ● Apply pattern recognition and sequencing to daily tasks. ● *Standard Met: 1.AP.A.01*

Unit 2: Sequences & Events (7 lessons)

Students explore how sequences and events form the foundation of programming. They will use drawing tools, event blocks, and sequence-building to design interactive animations and games. This unit also introduces debugging strategies and highlights real-world connections to responsible technology use and careers in computer science.

Objectives / Topics Covered	● Use drawing tools to create custom characters and backgrounds. ● Explain what events are and apply multiple event blocks in programs. ● Build sequences to model digital responsibilities. ● Apply debugging skills to identify and correct errors. ● Explore how computer science can solve community problems.
Lessons	**Drawing Tools: Nature Walk** ● Use drawing tools to create a nature-inspired scene. **Events** ● Explain events and use multiple event types in a single program. **Sequences: Digital Responsibilities** ● Create a program demonstrating responsible online behavior. **Hide and Seek Game** ● Use the hide block to make an interactive game.

	Introduction to the Wait Block - Add pauses to character actions using the wait block. **Introduction to Debugging** - Identify and correct mistakes in sequences. - *Standard Met: 1.AP.PD.03* **Careers in CS: Litter Free Communities** - Explore real-world applications of computer science in solving community problems.

Optional Unplugged Loops Activities

This optional unplugged unit gives students a hands-on way to practice programming concepts without using devices. By working together to move Scout through a maze using coding cards, students reinforce sequencing and looping skills in a fun, collaborative setting.

Objectives / Topics Covered	- Practice creating and following sequences of instructions without a screen. - Use loops to simplify repeated actions in a physical coding activity. - Build teamwork and communication skills while applying coding logic.
Lessons	**Coding Card Game: Loops** - Work together to guide Scout through a maze using cards that represent movement and repeat loops. **Coding Card Game: Loops 2** - Tackle a new maze challenge by building efficient sequences with loops to help Scout reach the goal.

Unit 3: Loops (3 lessons)

Students implement loops to simplify code, add rhythm, and create animations that repeat.

Objectives / Topics Covered	- Use different types of loops to repeat code. - Combine loops with motion and event blocks. - Develop games and animations with dynamic elements.
Lessons	**Introduction to Repeat Loops** - Use repeat loops to make a character perform an action multiple times in a row. **Loops: Catching Butterflies** - Animate butterflies using show/hide and repeat loops. - *Standard Met: 1.AP.C.01* **Forever Loop Dance Party** - Use the "repeat forever" loop to create a continuous dance party animation. - *Standard Met: 1.AP.C.01*

Unit 4: Message Events (5 lessons)

Students learn how to make characters interact and respond to each other using message events. They explore how messages can trigger sequences, change behavior, and allow for more complex and interactive programs..

Objectives / Topics Covered	- Program characters to communicate using message events. - Create interactive animations and stories with multiple characters. - Use events and variables to extend program functionality.
Lessons	**Introduction to Message Events** - Program a relay race using message events. **Message Events: Simon Says** - Create a “Simon Says” game where one character triggers multiple others.

	Create an Original Story Animation ● Students will be able to create a program to animate an original story. ● *Standards Met: 1.AP.PD.04, 1.AP.PD.01* **Speed Block: Bouncy Ball** ● Students will be able to use speed blocks and messages to program a character to move at different speeds. **Variables: Storing Data with Symbols** ● Students will be able to model the way programs store and manipulate data using variables. ● *Standards Met: 1.AP.V.01*

Unit 5: Pages & Grid (5 lessons)

In this unit, students build on their skills by learning how to use pages, grids, and special blocks to create interactive projects and games. They explore how programs can span multiple screens, move characters across grids, and include structured endings. These skills prepare students to design larger, more complex animations and activities that combine creativity with problem-solving.

Objectives / Topics Covered	● Using multiple pages in a program. ● Applying the "go to page" block to connect activities. ● Designing interactive games and greeting cards. ● Navigating with the grid to solve mazes. ● Programming endings with the "end" block.
Lessons	**Pages: Create a Tapping Game** ● Design an interactive game that switches between pages when a character is tapped. **Create a Mini Golf Game** ● Students will be able to use messages and loops to create a mini golf game. **Digital Greeting Card** ● Students will be able to create a digital greeting card with events and loops. **Grid: Solving Mazes** ● Design a maze and use grid-based movement to guide a character from start to finish. **End Block: Program a Race** ● Program a race between characters and use the "end" block to stop the animation at the finish line.

Unit 6: Culmination Projects (5 lessons)

In this unit, students apply the design process to create original projects demonstrating problem-solving, personal expression, and technical skill.

Objectives / Topics Covered	● Plan, create, and revise projects. ● Use sequences, events, loops, and variables. ● Model real-world or imaginative scenarios.
Lessons	**Practicing the Design Process** ● Use the design process to solve a simple real-world problem. ● *Standard Met: 1.AP.PD.04* **About Me Project (2 part lesson)** ● Design a program that shares personal characteristics, interests, and favorite things. **River Crossing Game (2 part lesson)** ● Create an obstacle game with difficulty controlled by speed blocks. ● *Standard Met: 1.AP.C.01*

Unit 7: Digital Literacy (9 lessons)

All lessons in this unit are required for full standards alignment. Students develop essential non-programming computer science skills, including research, data, networks, online safety, and understanding the impact of computing in everyday life.

Objectives / Topics Covered	• Use research tools to find and communicate information. • Collect, organize, and represent data visually. • Understand how computers store information as files. • Identify private and personal information. • Explain how networks and the Internet share information. • Practice positive and responsible online behavior. • Recognize the impact of computing on daily life and communities. • Compare different types of software and their purposes.
Lessons	**Guided Research (2 part lesson)** • Find information on a topic and create a visual program to present research. • *Standard Met: 1.AP.PD.02* **Basic Data and Programming** • Collect and present data through a visual program. • *Standards Met: 1.DA.CVT.01, 1.DA.IM.01* **Data Storage and Files Practice** • Learn how computers store information and explore how files hold data. • *Standard Met: 1.DA.S.01* **Keeping Information Safe** • Students will be able to identify private and personal information. • *Standards Met: 1.IC.SI.01, 1.NI.C.01* **Networks and the Internet** • Explain how networks work and how messages are shared online. • *Standard Met: 1.NI.NCO.01* **Positive Online Behavior** • Identify appropriate and inappropriate online behavior through an interactive program. • *Standards Met: 1.IC.SI.01, 1.IC.SLE.01, 1.NI.C.01* **Impacts of Computing: Our Community** • Identify examples of computing devices in daily life and explain how they've changed how people live, work, and play. • *Standard Met: 1.IC.C.01* **Types of Software** • Compare different types of software and select appropriate tools for specific tasks. • *Standard Met: 1.CS.D.01*

Maryland Computer Science 1st Grade Course Supplemental Materials

Resources	**Description**
Parent Welcome Letter (Spanish)	Send this letter home to introduce families to computer science.
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**.	