



Utah Computer Science 5th Grade Course Syllabus

One Year for Elementary School, 36 Hours

Course Overview and Goals

The Utah Computer Science 5th Grade Course introduces students to foundational programming concepts through 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. All lessons are labeled with the specific standards they address to support planning and compliance.

Standards Alignment Note: Standards addressed in each lesson are listed in parentheses after the lesson title. Standards may be reinforced across multiple lessons, and full coverage of state computer science standards is achieved through the course as a whole.

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.

Assessments: Some units conclude with a brief 5-question quiz designed to assess student understanding of the concepts and skills introduced in the unit. Quizzes can be assigned digitally, completed as a QuizBuzz activity, or printed for paper-based completion.

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/26173/overview>



A clickable PDF can be found at <https://codehs.com/UT-CSRoadmaps>

Course Breakdown

Optional Review

This optional review unit helps students refresh key computer science skills and prepare for the year ahead.

Objectives / Topics Covered	- Log in and navigate the Playground. - Review core computer science vocabulary and programming basics. - Practice using the coordinate plane for animation. - Customize sprites and scenes with drawing tools.
Lessons	Welcome to CodeHop! - Practice logging in and exploring the Playground before starting a full lesson. Introduction to Computer Science (5.AP.1) - Review basic computer science vocabulary and create a simple program. The Coordinate Plane (5.AP.1) - Use the coordinate plane to design an open-ended animation. Drawing Tools (5.AP.1) - Use drawing tools to create and personalize custom sprites and backgrounds.

Unit 1: Getting Started (2 lessons)

Students will explore the parts of computing systems. They'll also apply computational thinking to design creative solutions.

Objectives / Topics Covered	- Identify the parts of a computing system and how they function together. - Apply basic troubleshooting strategies to solve common technology issues. - Use computational thinking to solve real-world design challenges.
Lessons	Practicing Computing Systems (5.CS.1) - Identify key components of computing systems and apply troubleshooting strategies to resolve common technical issues. Computational Thinking: Design a Neighborhood (5.CT.1, 5.CT.2) - Use decomposition, pattern recognition, sequencing, and abstraction to design a neighborhood layout. Getting Started Quiz (5 questions)

Unit 2: Sequences & Events (5 lessons)

In this unit, students strengthen their understanding of sequences, events, and algorithm design by creating animations and interactive programs. They'll practice debugging, refine algorithms for efficiency, and explore how computer science is used in real-world contexts like marine biology and professional sports.

Objectives / Topics Covered	- Use sequences and events to control sprite behavior in programs. - Compare and improve algorithms based on clarity and efficiency. - Apply debugging strategies by decomposing complex programs. - Explore real-world uses of computer science, including storytelling through code. - Use drawing tools to create custom, animated characters.
Lessons	Drawing Tools: Sea Creatures (5.AP.1) - Use image editing tools to design and program animated sea creatures in a custom underwater scene. Events: Traveling with Scout (5.AP.1) - Use different types of events to animate Scout's journey through various locations. Compare and Refine Algorithms (5.AP.3) - Create multiple solutions to a problem, then compare and improve them to find the

	most efficient algorithm. Debugging: Make a Pizza (5.AP.4) - Decompose a program step by step to find and fix errors in a pizza-making animation. Careers in CS: Major League Baseball (5.IC.1) - Learn how coding is used in sports and retell important moments from an article using a timeline animation. Sequences & Events Quiz (5 questions)
--	--

Unit 3: Loops (2 lessons)

In this unit, students apply loops to create repeated designs using the pen tool, while also learning the importance of giving credit when using or remixing digital content. They'll combine creativity with responsible digital citizenship and efficient coding practices.

Objectives / Topics Covered	- Use loops to create repeating patterns and graphics with the pen tool. - Understand and apply the concept of attribution when using or sharing digital work. - Create original or remixed projects while practicing ethical digital behavior.
Lessons	Creating Turtle Graphics (5.AP.1) - Use repeat loops and the pen tool to draw looping patterns similar to turtle graphics. Giving Credit Through Attributions (5.IC.2) - Learn how to appropriately credit original creators when remixing programs or using shared images online. Loops Quiz (5 questions)

Unit 4: Conditionals & Operators (4 lessons)

In this unit, students explore how to use conditionals and operators to control program logic and add randomness. They'll build interactive projects like mazes, coin flips, and quests—practicing how to plan, decompose, and code using conditionals and logical expressions.

Objectives / Topics Covered	- Use if/then conditionals to control sprite behavior based on specific conditions. - Apply operators and variables to introduce randomness and comparisons. - Plan and break down complex projects into manageable steps. - Create interactive programs that respond dynamically to input or data.
Lessons	Conditionals: Mazes (5.AP.1) - Create a maze game using conditionals to guide a sprite through the correct path. Plan a Quest (2 classes 5.AP.3) - Decompose the steps needed to build a quest-style game, planning out conditionals, events, and logic. Operators: Coin Flip (5.AP.2) - Use variables and operators to create a coin-flipping program that simulates randomness and tracks outcomes. Conditionals & Operators Quiz (5 questions)

Unit 5: Variables & Lists (3 lessons)

In this unit, students deepen their understanding of how variables and lists can be used to manage and manipulate information in programs. Through creative projects like dance animations and shopping simulations, they apply variables, lists, and operators to create more dynamic and interactive experiences.

Objectives / Topics Covered	● Use variables to store and control values like speed and pitch. ● Create and manage lists to store multiple related items. ● Combine variables, lists, and operators to power interactive simulations. ● Build projects that reflect real-world problem-solving using data structures.
Lessons	Variables in Dance (5.AP.2) ● Use variables to change dance speed and pitch, creating a dynamic and musical animation. Lists: Shopping with Scout (2 classes 5.AP.2) ● Build a shopping simulator that uses lists to track items and variables and operators to calculate prices. Variables & Lists Quiz (5 questions)

Unit 6: Clones & Functions (5 lessons)

In this unit, students explore how to use clones to efficiently duplicate sprites and actions, particularly for game design. They'll also be introduced to the concepts of classes and objects in programming, using randomization to add variety and complexity to their projects.

Objectives / Topics Covered	● Use clones to duplicate sprites and streamline repetitive tasks in game development. ● Understand the value of clones for performance and efficiency in interactive programs. ● Explore foundational programming concepts like classes and objects. ● Use randomization to change object characteristics and enhance gameplay variety.
Lessons	Clones: Throwing Acorns Game (5.AP.1) ● Create a game where cloned acorns are thrown at targets, using events and movement to control gameplay. Clones in Games (2 classes 5.AP.1) ● Build an endless runner game using clones and explain how clones help manage repeated elements efficiently. Classes and Objects in Games (2 classes 5.AP.1) ● Learn the basics of classes and objects by creating a game where objects have random attributes like color, size, or speed. Clones & Functions Quiz (5 questions)

Unit 7: Culmination Projects (13 lessons)

In this final unit, students apply their coding, design, and problem-solving skills to complete original, real-world projects. From building games and designing homes with functions to creating apps and visualizing data, these projects offer students opportunities to showcase their mastery through creativity, and logic.

Objectives / Topics Covered	● Apply programming concepts like loops, conditionals, variables, and functions in original designs. ● Integrate math concepts like area and perimeter into a creative computer science project. ● Use the design thinking process to create a user-centered app. ● Follow the inquiry process to collect, analyze, and display data using graphs.
Lessons	Game Design Project (3 classes 5.AP.1, 5.AP.2) ● Design and build an original game using skills such as loops, conditionals, and variables. House Design with Area and Perimeter (2 classes 5.AP.1, 5.CT.2)

	• Use programming functions to design a digital house layout, applying area and perimeter calculations in a CS context. Design an App (3 classes 5.IC.1) • Follow the design thinking process to develop a prototype for an app that solves a real-world problem. Inquiry Project: Line Graph ((2 classes 5.DA.2, 5.DA.3) • Conduct an investigation and modify a program to display the results as a line graph.
--	---

Digital Literacy (3 lessons)

In this unit, students explore how computing innovations impact people and communities while learning how information travels across networks and how digital data is stored and managed. They also analyze data to draw conclusions and make informed decisions.

Objectives / Topics Covered	• Explore how computing innovations impact people and communities. • Understand how devices communicate across networks. • Explore how digital files are stored and managed. • Analyze data to draw conclusions and make predictions.
Lessons	Impacts of Computing: Innovation (5.IC.1) • Explore how computing innovations impact people and communities. Network Connections (5.NI.1, 5.NI.2) • Explore how devices communicate across wired and wireless networks. File Management and Data Exploration (5.DA.1) • Explore how digital files are stored, organized, and managed. Data Analysts (5.DA.2) • Analyze data to draw conclusions and make predictions. Digital Literacy Quiz (5 questions)

Optional: Artificial Intelligence (3 lessons)

In this optional unit, students explore how artificial intelligence compares to human thinking, how machine learning models are trained, and how AI products are designed and used responsibly. They also investigate how AI chatbots solve problems and communicate with users.

Objectives / Topics Covered	• Compare the strengths and limitations of humans and AI. • Explore how machine learning models use training data. • Analyze the benefits and challenges of AI products. • Investigate how AI chatbots answer questions and solve problems.
Lessons	Humans Vs. Machines (5.IC.1) • Compare the strengths and limitations of humans and AI. Machine Learning: Analyzing (5.DA.2) • Analyze how training data affects AI model predictions. AI Products: Ethical and Responsible Choices (5.IC.1) • Explore the benefits and challenges of AI products. Design Thinking: AI Chatbots (5.IC.1) • Explore how AI chatbots answer questions and support users. Artificial Intelligence Quiz (5 questions)

Utah Computer Science 5th 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 .	