

Let's Learn Computer Science 4



Created by Code.org on September 17, 2015; last modified by Tom Seaman on September 17, 2015

Unit Summary

This is Course 4 of a four-course set developed by Code.org intended to give elementary school students an engaging hands-on introduction to computer science. Course 4 is for students who have already completed courses 2 and 3, and delves deeper into the computer programming concepts introduced in those courses. Course 1 is aimed at early readers, generally in the K-2 grades. Teachers of students who can read and have no prior programming experience may elect to begin with Course 2. Course 2 is a prerequisite for Course 3, which gets into more advanced topics, such as nested loops and functions, and Course 4 goes even further into those topics. Each course employs a spiraling curriculum, where concepts are introduced and later revisited, delving deeper.

All four courses in the set utilize a blend of brief (1-2 minute) video introductions, teacher-led classroom activities, and hands-on exercises where students are expected to use the programming concepts to solve problems and create things, such as an interactive game, a drawing, or a story. Students can share their creations with their friends. Each course comprises 18 or so lessons, and each lesson requires roughly 25 to 45 minutes to complete. The lessons can be introduced by the teacher with the included teacher notes and prep videos. Also included are assessment plans and mapping to USA and international standards.

Course 4 exposes students to some important concepts in computer programming. The introductory videos explain in easy-to-understand language concepts like algorithms (“lists of steps to complete a task”), variables (“placeholders and containers to store values”), loops (“loops have a pre-specified beginning, end, and step interval”), parameters (“extra pieces of information for a function to customize it for a special need”), as well as if-else blocks and binary images. After each video, students progress through various challenges enabling them to roll up their sleeves and put to use their newfound knowledge of the subject just introduced. Students also have the opportunity to share their creations with friends.

At a Glance

- **Grade:** 3-5, 6-8
- **Subjects:** Science, Math, Arts, English Language Arts
- **Topics:** Computer Science, Engineering, Design
- **Higher-Order Thinking Skills:** Creativity, Collaboration, Persistence, Problem Solving
- **Key Learnings:** Computer programming concepts, vocabulary, collaboration
- **Content Type:** Unit Plan
- **Time Needed:** 18 lessons, each 25-45 minutes. May be done on consecutive days or over 18-week period
- **Prerequisites:** Ability to read
- **License:** Read about the license and what you can do with this material [here](#).

Learning Outcomes

- Students should gain an understanding of computer programming concepts, such variables, conditionals, nested loops, reusable functions, algorithms, parameters, if-else blocks, and binary images.
- Students should gain experience with computational thinking, creating algorithms to complete tasks, breaking down complex problems into smaller problems, and collaborating with others to achieve a better result than one could on one's own.
- Students should learn problem solving and techniques for persevering through difficult challenges.
- Students should learn that there are often multiple ways to solve a problem, and developing the most elegant solution sometimes takes creativity.

Things you Need

Computer with internet connection

This course requires students to use a computer or tablet with an internet connection. Students may work independently on the hands-on activities, but we recommend they work in pairs or threesomes to learn to collaborate to solve the challenges. The course utilizes YouTube to embed the instructional videos. If YouTube is blocked at your school, Code.org will attempt to play the video content through its own hosted non-YouTube video player. Additional IT help [here](#).

Link to Unit Plan

<https://studio.code.org/s/course4>

Students who have completed course 3 will take their skills further to solve more complex problems, ultimately creating an interactive game to share.