

# Grade 4 Technology Unit 4: Coding

Content Area: **Technology**  
Course(s): **Technology Grade 4**  
Time Period: **MP4**  
Length: **7 days**  
Status: **Published**

## **NJSLS - Computer Science and Design Thinking**

---

CS.3-5.8.1.5.AP.1	Compare and refine multiple algorithms for the same task and determine which is the most appropriate.
CS.3-5.8.1.5.AP.2	Create programs that use clearly named variables to store and modify data.
CS.3-5.8.1.5.AP.3	Create programs that include sequences, events, loops, and conditionals.
CS.3-5.8.1.5.AP.4	Break down problems into smaller, manageable sub-problems to facilitate program development.
CS.3-5.8.1.5.AP.5	Modify, remix, or incorporate pieces of existing programs into one's own work to add additional features or create a new program.
CS.3-5.8.1.5.AP.6	Develop programs using an iterative process, implement the program design, and test the program to ensure it works as intended.

## **Rationale and Transfer Goals**

---

Students will further develop their understanding of computer science using a unit of study developed by code.org. They will learn coding and problem solving through solving various coding puzzles, including how computers can complete some tasks significantly faster than people can without computers. This unit of study will help students understand the basics of how many digital apps, games and programs are created and could inspire them to further their knowledge of the subject as they get older. This unit will allow students to develop their critical thinking and problem solving skills, as well as their collaboration skills on challenging problems.

## **Enduring Understandings**

---

Computers can be used to quickly solve problems, but they require people to think creatively and be the driving force behind the problem solving.

## **Essential Questions**

---

How can I program a computer to complete a task?

How can coding affect the speed and efficiency with which the computer completes the task?

### **Content - What will students know?**

---

- Definition of Program and algorithm.
- Definition of loops.
- Definition of Functions.
- Definition of Conditionals.

### **Skills - What will students be able to do?**

---

- Follow an algorithm to create a pattern.
- Students will use a repeated action or group of actions to solve problems.
- Students will use functions as a shortcut to solve problems.
- Students will be able to create a card game using conditionals and will be able to determine when something should or should not happen as part of their program.

### **Activities - How will we teach the content and skills?**

---

- Large group instruction of graph paper programming, using 4x4 grids to create a pattern from a series of directions, followed by students creating their own algorithms to trade with classmates.
- Whole class demonstration and discussion of how an algorithm to guide a student around a classroom table could be shortened by looping certain directions. Students will then use loops to streamline their algorithms in code.org modules.
- Whole class discussion of a real life problem, creating a shortcut to make microwave popcorn, where the class will look at all the things a microwave does when it makes popcorn and how programmers use functions to make the instructions user friendly. They will then complete the Functions in Minecraft and Functions in Artist modules on code.org.
- Whole class discussion of conditional statements and how they are useful in programming. The class will then create and play a card game using conditionals and if/else statements, followed by the class

independently working through the Conditionals in Minecraft and Conditionals with the Farmer modules.

---

**Evidence/Assessments - How will we know what students have learned?**

---

- Evaluation of students completed patterns and algorithms.
- Evaluation of students' progress through Drawing With Loops and Fancy Shapes Using Nested Loops.
- Evaluation of students' progress through Functions in Minecraft and Functions with Artist modules.
- Evaluation of students' progress through the Conditionals in Minecraft and Conditionals with the Farmer modules.

**Spiraling for Mastery**

Content or Skill for this Unit	Spiral Focus from Previous Unit	Instructional Activity
Conditionals	How to combine movement and actions in algorithms	Whole class discussion of conditional statements and how they are useful in programming. The class will then create and play a card game using conditionals and if/else statements, followed by the class independently working through the Bee: Conditionals module.

---

**Key Resources**

[www.code.org](http://www.code.org) course E

4x4 graph paper

WRK.9.2.5.CAP.1	Evaluate personal likes and dislikes and identify careers that might be suited to personal likes.
WRK.9.2.5.CAP.4	Explain the reasons why some jobs and careers require specific training, skills, and certification (e.g., life guards, child care, medicine, education) and examples of these requirements.

## **Career Readiness, Life Literacies, & Key Skills**

---

TECH.9.4.5.CT.3	Describe how digital tools and technology may be used to solve problems.
-----------------	--------------------------------------------------------------------------

## **Interdisciplinary Connections/Companion Standards**

---

Literacy and language arts in the technology context: writing, programming, word processing, and creativity with language

Science: understanding of computer components, operations of touchscreens and other user devices

Social Studies: Computers in the context of society; our relationships to computers as a tool

Health: Limits to screen time and healthy relationships with technology, online technologies

STEM: Applied science and math in the context of game applications and coding

SCI.3-5-ETS1-1	Define a simple design problem reflecting a need or a want that includes specified criteria for success and constraints on materials, time, or cost.
----------------	------------------------------------------------------------------------------------------------------------------------------------------------------