

Programming Karel

Content Area: **Math**
Course(s): **Introduction to Computer Science**
Time Period: **Marking Period 2**
Length: **5 - 6 Weeks**
Status: **Published**

Unit Overview

This unit introduces students to programming the motion of Karel the Dog. Students will learn and understand how to use many different tools in programming to help Karel perform very specific tasks.

Enduring Understandings

Objectives / Topics Covered

- Commands
- Defining vs. Calling Methods
- Designing methods
- Program entry points
- Control flow
- Looping
- Conditionals
- Classes

- Commenting code
- Preconditions and Postconditions
- Top Down Design

Essential Questions

- How do you break a bigger task into smaller (more easily programmed) tasks?
- How can smaller tasks be used to solve more complex problems?
- How can your coding be done in a way that another reader understands the process? (Abstraction)
- Can you give specific instructions (algorithms) to solve a specific problem?

New Jersey Student Learning Standards (No CCS)

8.1.8.AP.3: Design and iteratively develop programs that combine control structures, including nested loops and compound conditionals. Programs use procedures to organize code and hide implementation details. Procedures can be repurposed in new programs. Defining parameters for procedures can generalize behavior and increase reusability.

8.1.8.AP.4: Decompose problems and sub-problems into parts to facilitate the design, implementation, and review of programs.

8.1.8.AP.6: Refine a solution that meets users' needs by incorporating feedback from team members and users.

8.1.8.AP.7: Design programs, incorporating existing code, media, and libraries, and give attribution.

8.1.8.AP.8: Systematically test and refine programs using a range of test cases and users.

8.1.8.AP.9: Document programs in order to make them easier to follow, test, and debug.

Technology Standards

TECH.8.1.12	Educational Technology: All students will use digital tools to access, manage, evaluate, and synthesize information in order to solve problems individually and collaborate and to create and communicate knowledge.
TECH.8.1.12.A.3	Collaborate in online courses, learning communities, social networks or virtual worlds to discuss a resolution to a problem or issue.
TECH.8.1.12.A.CS1	Understand and use technology systems.
TECH.8.1.12.A.CS2	Select and use applications effectively and productively.
TECH.8.1.12.B.2	Apply previous content knowledge by creating and piloting a digital learning game or tutorial.
TECH.8.1.12.B.CS1	Apply existing knowledge to generate new ideas, products, or processes.
TECH.8.1.12.B.CS2	Create original works as a means of personal or group expression.
TECH.8.1.12.C.CS4	Contribute to project teams to produce original works or solve problems.
TECH.8.2.12.C.CS1	The attributes of design.
TECH.8.2.12.E.1	Demonstrate an understanding of the problem-solving capacity of computers in our world.
TECH.8.2.12.E.3	Use a programming language to solve problems or accomplish a task (e.g., robotic functions, website designs, applications, and games).
TECH.8.2.12.E.CS1	Computational thinking and computer programming as tools used in design and engineering.

21st Century Themes/Careers

List specific standards that are relevant
No general statements

CAEP.9.2.12.C.3	Identify transferable career skills and design alternate career plans.
CAEP.9.2.12.C.5	Research career opportunities in the United States and abroad that require knowledge of world languages and diverse cultures.

Instructional Strategies & Learning Activities

- 26 Karel programming exercises in total
- Program-specific tasks for Karel the Dog
 - Example Exercise: Pyramid of Karel Write a program to have Karel build a pyramid. There should be three balls on the first row, two in the second row, and one in the third row.
- Teach Karel new commands like turnRight() or makePancakes()
 - Example Exercise: Pancakes Karel is the waiter. He needs to deliver a stack of pancakes to the guests on the 2nd, 4th, and 6th avenue. Each stack of pancakes should have three pancakes. Create a method called

makePancakes() to help Karel solve this problem.

- Solve large Karel problems by breaking them down into smaller, more manageable problems using Top Down Design
 - Example Exercise: The Two Towers In this program, Karel should build two towers of tennis balls. Each tower should be 3 tennis balls high. At the end, Karel should end up on top of the second tower, facing East.
- Using control structures and conditionals to solve general problems
 - Example Exercise: Random Hurdles Write a program that has Karel run to the other side of first street, jumping over all of the hurdles. However, the hurdles can be in random locations. The world is fourteen avenues long.

5 Karel challenges to tie everything learned in the Karel module together

○Example Exercise: Super Cleanup Karel Karel's world is a complete mess. There are tennis balls all over the place, and you need to clean them up. Karel will start in the bottom left corner of the world facing east, and should clean up all of the tennis balls in the world. This program should be general enough to work on any size world with tennis balls in any locations.

Formative Assessments

Students will take short quizzes each Friday (a couple of MC, one or two short answer and a programming task).

Closure check in quizzes (1 - 2 questions) after videos / lessons.

Summative Assessment

End of unit MC tests from Code HS.

Larger programming challenges.