



# CodeHS

## Intro to Programming with Karel the Dog Syllabus

1 quarter for Middle or High School (30 contact hours)

### Course Overview and Goals

The CodeHS Introduction to Programming with Karel the Dog course teaches students the basics of programming by giving commands to a computer just like you give commands to a dog. Karel is a dog that lives in a grid world and can be instructed to move around and pick up and put down tennis balls. Students will learn JavaScript commands, functions, and control structures by solving puzzles and writing creative programs for Karel to follow.

**Learning Environment:** The course utilizes a blended classroom approach. The content is a mix of web-based and physical activities. Students will write and run code in the browser, and engage in in-person collaborative exercises with classmates. Teachers utilize tools and resources provided by CodeHS to leverage time in the classroom and give focused 1-on-1 attention to students.

**Programming Environment:** Students write and run programs in the browser using the CodeHS online editor. Students will be able to write both text based and block based programs in Karel.

**Quizzes:** Each lesson includes at least one formative short multiple choice quiz. At the end of each unit, students take a summative multiple choice unit quiz that assesses their knowledge of the concepts covered in the unit.

**Prerequisites:** The Introduction to Programming with Karel the Dog course is designed for complete beginners with no previous background in computer science. The course is highly visual, dynamic, and interactive, making it engaging for those new to computer science.

**More information:** Browse the content of this course at <https://codehs.com/course/23238>

## Course Breakdown

### Unit 1: Introduction to Programming (1 week/5 hours)

|                             |  |
|-----------------------------|--|
| Objectives / Topics Covered | <ul style="list-style-type: none"><li>● What is programming?</li><li>● Commands</li><li>● Introduction to functions</li><li>● Debugging</li></ul>  |
| Example Assignments / Labs  | <ul style="list-style-type: none"><li>● Commands<ul style="list-style-type: none"><li>○ Program-specific tasks for Karel the Dog</li><li>○ Example Exercise: Pyramid of Karel<br/>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.</li></ul></li><li>● Functions<ul style="list-style-type: none"><li>○ Teach Karel <code>turnRight()</code></li><li>○ Example Exercise: Fireman Karel<br/>There is a fire, and Karel needs to slide down a fireman's pole. Define and use the <code>turnRight()</code> function to help Karel get to the ground.</li></ul></li><li>● Debugging<ul style="list-style-type: none"><li>○ Students reflect on their debugging experience</li></ul></li></ul> |

### Unit 2: Functions (1 week/4 hours)

|                             |   |
|-----------------------------|---|
| Objectives / Topics Covered | <ul style="list-style-type: none"><li>● Functions</li><li>● The Main function</li><li>● Top Down Design and Decomposition</li></ul>   |
| Example Assignments / Labs  | <ul style="list-style-type: none"><li>● Functions<ul style="list-style-type: none"><li>○ Teach Karel new commands like <code>turnRight()</code> or <code>makePancakes()</code></li><li>○ Example Exercise: Pancakes<br/>Commands<ul style="list-style-type: none"><li>■ Program-specific tasks for Karel the Dog</li><li>■ Example Exercise: Pyramid of Karel<br/>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.</li></ul></li></ul></li><li>● Top Down Design<ul style="list-style-type: none"><li>○ Solve large Karel problems by breaking them down into smaller, more manageable problems</li><li>○ Example Exercise: The Two Towers<br/>In this program, Karel should build two towers of tennis balls. Each tower should be 3 tennis balls high.</li></ul></li></ul> |

|  |  |
|--|--|
|  | At the end, Karel should end up on top of the second tower, facing East. |
|--|--|

### Unit 3: Super Karel and For Loops (1 week/4 hours)

|                             |  |
|-----------------------------|--|
| Objectives / Topics Covered | <ul style="list-style-type: none"> <li>● Commenting Your Code</li> <li>● Super Karel</li> <li>● For Loops</li> </ul>   |
| Example Assignments / Labs  | <ul style="list-style-type: none"> <li>● Commenting Your Code <ul style="list-style-type: none"> <li>○ Commenting is important to make sure your code is understandable to yourself and others. <ul style="list-style-type: none"> <li>■ Example Exercise: The Two Towers + Comments<br/>Take your Two Towers program from earlier, and comment each of the functions to explain what your program is doing.</li> </ul> </li> </ul> </li> <li>● Super Karel <ul style="list-style-type: none"> <li>○ Super Karel knows how to <code>turnRight()</code> and <code>turnAround()</code>, so you don't need to define these functions before you call them</li> <li>○ Example Exercise: The Two Towers + SuperKarel<br/>Take your Two Towers program from earlier, and make it work with SuperKarel.<br/>This means you can get rid of your definition for <code>turnRight</code>, because SuperKarel already knows <code>turnRight</code>!</li> </ul> </li> <li>● For Loops <ul style="list-style-type: none"> <li>○ For loops execute the code inside the loop a set number of times.</li> <li>○ Example Exercise: Dizzy Karel<br/>Write a program to have Karel spin in place 8 times.<br/>This means Karel is going to need to turn left 32 times.<br/>You must use a for loop in this program.</li> </ul> </li> </ul> |

### Unit 4: If Statements (0.5 week/2 hours)

|                             |   |
|-----------------------------|---|
| Objectives / Topics Covered | <ul style="list-style-type: none"> <li>● If Statements</li> <li>● If/Else Statements</li> <li>● Debugging strategies</li> </ul>   |
| Example Assignments / Labs  | <ul style="list-style-type: none"> <li>● If statements <ul style="list-style-type: none"> <li>○ If statements will execute code only if certain conditions are met</li> </ul> </li> </ul> |

|  |  |
|--|--|
|  | <ul style="list-style-type: none"> <li> <ul style="list-style-type: none"> <li> <ul style="list-style-type: none"> <li>■ Example Exercise: Is there a ball?<br/>If there is a ball in the first spot, move once. Otherwise, put a ball down and then move once. Karel should end up on the second column in the first row, and there should be a ball at 1, 1.</li> </ul> </li> </ul> </li> <li>● If/Else statements <ul style="list-style-type: none"> <li>○ The if/else statement executes a block of code if a specified condition is true. If the condition is false, another block of code can be executed.</li> <li>○ Example Exercise: Right Side Up<br/>Karel got spun around and is facing the wrong direction.<br/>Karel could either be facing south or west, and you need to get Karel back to facing east. You must use an if/else statement here.</li> </ul> </li> </ul> |
|--|--|

### Unit 5: While Loops (0.5 week/2 hours)

|                             |  |
|-----------------------------|--|
| Objectives / Topics Covered | <ul style="list-style-type: none"> <li>● While Loops</li> </ul>  |
| Example Assignments / Labs  | <ul style="list-style-type: none"> <li>● While Loops <ul style="list-style-type: none"> <li>○ A while loop allows code to be executed repeatedly based on a given Boolean condition <ul style="list-style-type: none"> <li>■ Example Exercise: Big Tower<br/>You need to make a tall tower (all the way to the top) of tennis balls in the column Karel is standing in at the start. Karel could be in any column, and the world's height could be any size. Your solution should include more than one helper function.<br/>Karel can start facing any direction, so you need to make sure to orient Karel to the North.</li> </ul> </li> </ul> </li> </ul> |

### Unit 6: Control Structures and Ultra Karel (1 week/4 hours)

|                             |   |
|-----------------------------|---|
| Objectives / Topics Covered | <ul style="list-style-type: none"> <li>● Control Structures</li> <li>● Indenting Code</li> <li>● Ultra Karel</li> </ul>   |
| Example Assignments / Labs  | <ul style="list-style-type: none"> <li>● Control Structures <ul style="list-style-type: none"> <li>○ Students learn how to determine which control structure is appropriate for a given task <ul style="list-style-type: none"> <li>■ Example Exercise: Decorate the Fence<br/>Karel is getting ready for Halloween by putting</li> </ul> </li> </ul> </li> </ul> |

|  |  |
|--|--|
|  | <p>up jack-o-lantern lights. Help Karel by finding the wall to string lights on and putting down lights along the wall. However, Karel doesn't want lights blocking the doors and windows! Make sure that you only put lights where there is a wall.</p> <p>Karel starts in the bottom left corner facing East. Karel should end at the top of the wall, facing North.</p> <ul style="list-style-type: none"> <li>● Indenting Code <ul style="list-style-type: none"> <li>○ Indenting Code makes programs more readable and helps you avoid error. <ul style="list-style-type: none"> <li>■ Example exercise: Staircase<br/>This program builds a staircase in any sized square world, but the indenting is all wrong. Your job is to fix it, so the indentation is correct. Remember, push tab to indent once to the right and delete/backspace to remove a tab.</li> </ul> </li> </ul> </li> <li>● Ultra Karel <ul style="list-style-type: none"> <li>○ Ultra Karel knows how to paint with the <code>paint</code> command. <ul style="list-style-type: none"> <li>■ Example exercise: Checkerboard Karel<br/>Karel creates a checkerboard by painting alternating squares black and red.</li> </ul> </li> </ul> </li> </ul> |
|--|--|

### Unit 7: Karel Challenges (1 week/5 hours)

|                                    |  |
|------------------------------------|--|
| <p>Objectives / Topics Covered</p> | <ul style="list-style-type: none"> <li>● Commands</li> <li>● Defining versus Calling Methods</li> <li>● Designing methods</li> <li>● Control flow</li> <li>● Looping</li> <li>● Conditionals</li> <li>● Commenting code</li> <li>● Preconditions and Postconditions</li> <li>● Top Down Design</li> </ul>  |
| <p>Example Assignments / Labs</p>  | <ul style="list-style-type: none"> <li>● Challenges <ul style="list-style-type: none"> <li>○ Students use all of the skills learned in the course to solve complex puzzles and challenges. <ul style="list-style-type: none"> <li>■ Example exercise: Fetch<br/>Karel starts on the floor and sees a ball up on a shelf. Karel needs to go fetch the ball and bring it back to the starting position.</li> </ul> </li> </ul> </li> </ul> |

**Unit 8: Final Project (0.5 weeks/2 hour)**

|                             |  |
|-----------------------------|--|
| Objectives / Topics Covered | Final Project  |
| Assignments / Labs          | <ul style="list-style-type: none"><li>• Students brainstorm, discuss, and pseudocode to create a fully functional Karel program that solves a puzzle, creates a picture, or completes some other challenge of their choosing</li></ul> |

**Optional Supplemental Materials**

These supplemental materials should be used following the Prerequisite Units mentioned:

| <b>Supplementary Units</b>       | <b>Prerequisite/Recommended Unit(s)</b>     | <b># of activities</b> |
|----------------------------------|---|------------------------|
| Final Karel Multiple Choice Quiz | Completing the Karel course                 | 1                      |
| Extra Karel Activities           | Varies                                      | 23                     |
| Karel Syntax Introduction        | None  | 20                     |
| JavaScript Basics                | Completing all Karel modules is recommended | 39                     |