



KAREL JR 1  
STUDENT JOURNAL

REVISED JUNE 25, 2016

NAME	
DATE STARTED	DATE COMPLETED
SCHOOL, CLASS, PERIOD	



## TABLE OF CONTENTS:

WELCOME TO YOUR JOURNAL	4
SECTION 1: MANUAL MODE, BASIC FEATURES, CREATIVE SUITE	5
SECTION 2: BASIC COMMANDS, SYNTAX	9
SECTION 3: REPEAT LOOPS	13
SECTION 4: REPEAT LOOPS EMBEDDED IN PROGRAMS	17
SECTION 5: MULTIPLE LOOPS AND NESTED LOOPS	21
REVIEW YOUR PROGRESS	25
LIST OF BASIC COMMANDS AND KEYWORDS	26
LIST OF KEY VOCABULARY	27
FILE LOG: GAMES I HAVE CREATED	28
DESIGN TEMPLATE	29
NOTES	30

General Website: <https://nclab.com/>

Karel Gallery : <https://nclab.com/karel-gallery/>

Desktop (needs login information) <https://desktop.nclab.com/>

*Keep your name and password in a safe place.*

## WELCOME TO YOUR JOURNAL

### I'm already learning on-line. Why do I need a journal?

Good question. There are many benefits to journaling. In a nutshell, journaling will empower you to:

1. **Remember better:** When you write something down, you are **exporting** knowledge from your brain. This makes the brain think about what the learning that it has just **imported**.
2. **Create your own reference book:** Wow- everything you have learned will be in one place - a treasured notebook that you can read anytime and anywhere, customized by your own experience. Years later, you can show your grandchildren how you learned coding. In the meantime, your trusty journal will be at your side whenever you need to look something up that you have forgotten.
3. **Make connections:** By going back and forth in your notebook, you are connecting all the pieces. A journal also provides a place to connect the coding to observations you make in **real life**. After all, why do we program? To solve programs in the real world! Maybe you will have a brilliant idea that will be the next app. Jot down all your ideas.
4. **Keep a record of your work:** You will be creating a lot of files, games and puzzles. At the back of the journal is a log page where you can keep a record of your creations, just in case some event vaporizes your work. It can happen. (Reminder: always back up your files)
5. **Use your notes to collaborate with others:** Share what you write and create. It's amazing how other people see things both the same way you do, and differently. You will fill holes, catch mistakes, improve each other's work, and maybe even work on a project together.

### OK, so maybe this extra effort will pay off. What's in this journal?

Each course journal is set up the same way. The journal is divided into sections that match those in the on-line course. Each section has:

1. One or two **review pages** with questions or activities to help you remember what you have learned. There will always be an open box for your own notes.
2. A **bulletin board** where you can post real life examples: paste in pictures, sticky notes, and scribbles. There are ideas and suggestions at the top of each bulletin board.
3. A **planning page** for your end-of-section project.

The back of the journal contains a **glossary**, a **record page** for your files, and a **design template** that can be copied to work on games.

Slow down, journal, talk with people, and sketch ideas. We hope you will develop a deeper understanding of what coding is all about, and discover the thrill of having a computer or machine carry out a program that you have written.

Happy coding!

## SECTION 1: MANUAL MODE, BASIC FEATURES, CREATIVE SUITE

In Section 1, you learn how to use manual mode controls. You can control Karel's actions using the keyboard or buttons on the screen.

Write down the five keywords (commands) that you learned in Section 1 and note the key or button used for the tasks. What exactly does the command do? The first one has been done for you.

KEYWORD (COMMAND)	KEYBOARD KEY	SCREEN BUTTON	FUNCTION
go	Up arrow 		Move Karel one step forward

This section also teaches you to be aware of how efficiently your program works. The program is checked for the number of operations, the number of steps, and amount of time taken. Why would each of these factors be important in programming?

FACTOR	WHAT IT MEASURES	WHY IT IS IMPORTANT
Number of operations		
Number of steps		
Amount of time		

**SECTION 1 NOTES**

*Use this space to write your own notes, questions, and problems.*


**QUESTIONS**

Many of these levels restricted the number of steps you could take. Did you plan ahead, or just keep trying until you were successful? How can you plan the number of steps to stay less than or equal to the maximum allowed?

--

Discuss at least two different pathways through the maze to complete Level 1.6 "Fire!" Is there any advantage to using one rather than the other?

--

What pattern was needed to complete Level 1.7 "Flowers" within 13 steps? Would you have chosen this pattern without the fences to guide you?

--

## SECTION 1 BULLETIN BOARD

<i>This is a page to post ideas, pictures, sticky notes, drawings</i>	<i>Where have you seen robots in action?</i>	<i>If you could have a robot do your chores, what directions would you give it?</i>

## SECTION 1 PROJECT

Use this page to develop ideas for your game for Karel. You do not need to write any code for manual mode. Instead, you can draw the solution paths through the maze. In manual mode, you might want to restrict the number of steps or operations to make the maze more challenging. **Just make quick notes here: you can make more detailed notes and description in Designer Mode using Edit Game.**

Game Name:				Date:			
Maze Sketch (15 x 12)						Key:	
Storyline:							
Karel's goals:							
Number of steps:							
Number of operations:				Keywords:			
Any special challenges:							

## SECTION 2: BASIC COMMANDS AND SYNTAX

In Section 2, you are typing the same keyword/commands that you used in Section 1.

**Programming terms:** command, operation, lines of code

**Command words:** go, left, right, get, put

These typed commands replace the button and keyboard controls used in Section 1.

This section also teaches you to write one command per line, and that each commands start at the beginning of line. Rewrite the following lines of code correctly.

INCORRECT CODE	CORRECTED	WHAT ERRORS WERE MADE?
<pre>go go get go right put</pre>		
<pre>go     get go   left go go</pre>		

Write the code needed for Karel to move forward two steps, turn right, move forward one step, pick up an object, turn left, move forward three steps, put the object into a container, turn right, move two steps. What will this look like when you run the program? How many operations are there? \_\_\_\_\_  
How many steps did Karel take? \_\_\_\_\_

Program



**SECTION 2 NOTES**

*Use this space to write your own notes, questions, and problems.*


**QUESTIONS**

What happens when you give a robot a command that is not correct? Give an example.

--

Think of a simple procedure you do every day, like putting your books away, eating lunch, getting dressed. How could you write code for such a procedure using `go`, `left`, `right`, `get`, `put`?

--

In Karel, what are SHIFT-ENTER and the eraser button used for?

--

## SECTION 2 BULLETIN BOARD

<i>This is a page to post ideas, pictures, sticky notes, drawings</i>	<i>Where have you seen robots in action?</i>	<i>What are your favorite themes, objects, and containers?</i>



### SECTION 3: REPEAT LOOPS

In this section, you learn how to use the `repeat` loop. You also know that the `repeat` command must be followed by a number, the body of the loop is indented, and the loop can repeat one or more commands.

Let's review vocabulary. Match each term to the correct definition.

TERM
1. body
2. algorithm
3. syntax error
4. loop
5. logical error
6. computer program
7. syntax
8. <code>repeat</code>

DEFINITION
a. a command that tells how many times a set of commands will be repeated
b. a mistake in spelling, operators, indentations, spaces
c. An algorithm written using a programming language
d. A set of commands repeated a given number of times
e. the way a command line is written
f. a mistake in an algorithm
g. a series of logical steps that leads to the solution of a task
h. contains the commands to be repeated

Looking for patterns in programs takes study and planning. What repeated patterns do you see in this picture?




**SECTION 3 NOTES**

*Use this space to write your own notes, questions, and problems.*


**QUESTIONS**

What syntax do you use when writing `repeat` loops?

--

Cooking often requires repeated procedures: putting cookie dough in rows on a baking sheet is one example. Can you think of others?

--

<code>go</code>	vs.	<code>repeat 2</code>	Both use two lines. When would a <code>repeat 2</code> loop be useful?
<code>go</code>		<code>go</code>	

--

## SECTION 3 BULLETIN BOARD

<i>This is a page to post ideas, pictures, sticky notes, drawings</i>	<i>Where have you seen robots in action?</i>	<i>What real life situations are like mazes?</i>



## SECTION 4: REPEAT LOOPS EMBEDDED IN PROGRAMS

In Section 4, you learn how to figure out the body of a loop with certainty, and write commands before and after a loop. You also know that to put commands after a loop, your indentation must be canceled.

Compare these two examples with different indentations. Draw Karel's path for each one.

Program 1
<pre>repeat 3   go   left   go   right go</pre>



Program 2
<pre>repeat 3   go   left   go   right   go</pre>



Write a general rule for indentation with loops. Explain how to start the loop, and how to end it.


**SECTION 4 NOTES**

*Use this space to write your own notes, questions, and problems.*


**QUESTIONS**

In 4.7, there were several possible solutions. The program can be written in 17, 16, or 15 lines. Compare the solutions (discuss with a friend).

What is being done to reduce the number of lines?
Is the program more effective when it is shorter?

## SECTION 4 BULLETIN BOARD

<i>This is a page to post ideas, pictures, sticky notes, drawings</i>	<i>Video games are complex programs. Analyze a favorite game for patterns.</i>	<i>Think of a quest in a video game, where you travel through a scene and do several tasks. How would this look as a program that contained several repeat loops?</i>

## SECTION 4 PROJECT

Use this page to develop ideas for your game for Karel. The person playing your game will need to write code to solve the game. This time, **embed one or more** repeat **loops in a larger program**. For your notes, you can draw the solution paths through the maze, and write the correct code to the left. **Just make quick notes here: you can make more detailed notes and description in Designer Mode using Edit Game.**

Game Name:	Date:
Program:	<p data-bbox="669 527 932 558">Maze Sketch (15 x 12)</p> 
Storyline:	
Karel's goals:	
Number of steps:	
Number of operations:	Keywords:
Any special challenges:	



## SECTION 5 NOTES

*Use this space to write your own notes, questions, and problems.*

## QUESTIONS

How are multiplication and division similar to nested loops?

A gardener plants a row of corn. She is planning 10 plants for the row. She puts three seeds in each hole, hoping that at least one will germinate and grow into a plant (in real life, these would be spaced closely, but for now, take a step each time you “plant” a seed). She spaces the holes apart by two steps. Use `go`, `left`, `right`, `put` to write a program to do this task.

## SECTION 5 BULLETIN BOARD

<i>This is a page to post ideas, pictures, sticky notes, drawings</i>	<i>Repeated patterns are often used in agriculture.</i>	<i>Thinking of repeated patterns in tasks, spacing, and time.</i>



## REVIEW YOUR PROGRESS

This is the final Section of Karel Jr 1. Reflect on what you have learned so far.

Rate yourself C, B, or A:

- C if you could use this skill any time and could coach someone else;
- B if you have a good understanding but need more practice, and
- A if you feel that you are unsure of yourself and need teaching or coaching.

SKILL OR CONCEPT	C	B	A
Movement controls; basic commands <code>go</code> , <code>left</code> , <code>right</code> , <code>get</code> , <code>put</code>			
<code>repeat</code> loops; <code>repeat</code> loops with code before or after the loop.			
Nested loops			
Creating a game in Creative Suite			

Now, set some learning goals based on your self-evaluation. Don't worry if you aren't an expert in everything yet!

<b>RETAKE CERTAIN LEVELS</b> <b>FIND A COACH</b> <b>REVIEW AND DISCUSS NOTES</b> <b>PRACTICE</b>	<b>PRACTICE</b> <b>REVIEW AND DISCUSS NOTES</b> <b>CREATE</b>	<b>READY FOR THE NEXT COURSE</b> <b>CREATE</b> <b>COACH</b>

## LIST OF BASIC COMMANDS AND KEYWORDS – KAREL JR 1

Command words: `go`, `left`, `right`, `get`, `put`

Directional commands (`go`, `left`, `right`) are always from the robot's point of view.

`go` advances the robot one step.

`left` turns the robot to its left.

`right` turns the robot to its right.

Retrieving and placing objects (`get`, `put`)

`get` picks up an object

`put` places an object

Loops

`repeat x`, where `x` = the number of times the command is to be repeated.

## LIST OF KEY VOCABULARY KAREL JR 1 (IN ORDER OF APPEARANCE)

**Command words:** `go`, `left`, `right`, `get`, `put`. These words tell Karel what to do.

**Home** is the destination square, marked by red diagonal stripes which change to green when Karel approaches the square. The word `home` is also used in conjunction with commands.

**Max** may refer to maximum number of steps, operations, or programming lines.

**Steps** are the number of squares that Karel moves. The shoe icon  counts the number of steps.

**Operations** are anything that Karel does: move, turn, pick up or put down objects. The computer icon  counts the number of operations.

**Objects** are items placed in the maze. (The word “object” can have other connotations in programming that are not used here).

`repeat` is written on its own line as `repeat x`, where `x` = the number of times the command is to be repeated.

**Body:** the body contains the commands to be repeated. The commands are written on the lines following the `repeat` command, indented two spaces.

**Loop:** A set of commands repeated a given number of times.

**Nested loop:** A loop that is within another loop.

This is a good time to introduce some of the terms used in programming. Refer to the online textbook under Section 5 Programming for details.

**Algorithm:** a series of logical steps that leads to the solution of a task. Students may be familiar with algorithms used in operations such as subtraction and long division.

**Logical error:** a mistake in an algorithm. Planning helps reduce the number of errors.

**Computer Program:** An algorithm written using a programming language.

**Syntax:** the way a command line is written.

**Syntax error:** a mistake in spelling, operators, indentations, spaces

## FILE LOG: GAMES I HAVE CREATED

FILE NAME AND LOCATION	DATE	DESCRIPTION	NOTES
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			



## NOTES