Ozobot Editor—Lesson: Array Primer—Teacher's Guide

Grades 6 – 12

by Richard Born, Associate Professor Emeritus, Northern Illinois University

Introduction

If you haven't already done so, you should read the *Ozobot Editor Array Primer Student Guide*. Doing so will familiarize you with the lesson details that your students will experience as they are introduced to the concept of arrays in computer science.

This lesson uses several computer science programming concepts:

- Creation and manipulation of a one-dimensional array of integers using "List" blocks
- The use of variables to hold the values of program constants
- The use of loops to handle tasks that must be repeated several times
- Variable names that support self-documentation

Applicable CSTA (Computer Science Teachers Association) Standards

2-CS-02	Design projects that combine hardware and software components to collect and exchange data.
2-AP-11	Create clearly named variables that represent different data types and perform operations on their values.
3A-DA-11	Create interactive data visualizations using software tools to help others better understand real-world phenomena.
3B-AP-14	Construct solutions to problems using student-created components, such as procedures, modules and/or objects.

Classroom Implementation of this Lesson

Provide each student or lab group with a printed copy of the *Ozobot Editor Array Primer Student Guide* and *Ozobot Editor Array Primer Ozomap*. The student guide should be printed in color, but the Ozomap can be printed in black and white if preferred. All programs have been tested and work well with either Evo or Ari.

If time is an issue, you could provide students with the share codes for each of the eleven programs. These codes can be found in the **TXT** file *Ozobot Editor Array Primer Share Codes*. If time is not an issue, it is really quite instructive for students to create the first program directly from blocks and then modify the program progressively as they work through the eleven lessons.

Video

There is a video included with this lesson that shows Ari moving on the Ozomap for Program 11 on sorting an array. You may want to show this to your students to give them a feel for how Ozobot responds by moving and speaking the values of array elements.

Answers to Program Questions

Program 1: Element values: 0, 0, 0, 0, 0, 0, 0, 0, 0, 0.

Program 2: All the Same to Me – Element values: 5, 5, 5, 5, 5, 5, 5, 5, 5, 5.

Program 3: To Err is Human – Any time that an array references an index that is out-of-bounds (*i.e.,* less than zero or greater than *length – 1*) the program will stop with an index out of bounds error.

Program 4: The Count – Expected array element values: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Program 5: Counting by 5's – The expected array element values: 0, 5, 10, 15, 20, 25, 30, 35, 40, 45

Program 6: All Squared Up – Expected array element values: 0, 1, 4, 9, 16, 25, 36, 49, 64, 81.

Program 7: Blast off! – In Figure 8, array elements are set to the value *i*, whereas in Figu, they are set to the value *i – 1*. Expected array element values: 9, 8, 7, 6, 5, 4, 3, 2, 1, 0.

Program 8: Getting Dicey – The "set #" block in Figure 10 is setting the value to a random integer between 1 and 6 rather than to the constant 5, as in the program of Figure 5. The element values will be random numbers between 1 and 6, as in rolling a single ordinary 6-sided die. The random numbers will be different each time the program is run.

Program 9: One by One – Expected array element values: 114, 199, 36, 127, -89, 71, 107, -37, 45, 5.

Program 10: An Improvement – The most significant advantage of using the "length of" block is that in the event that we want to change the length of the array, we only need to adjust the value in the "create list" block. No adjustments need to be made elsewhere in the program.

Program 11: Sort of Interesting

Array B ascending: -89, -37, 5, 36, 45, 71, 107, 114, 127, 199

Array B descending: 199, 127, 114, 107, 71, 45, 36, 5, -37, -89

ArrayA remains as it was initialized with all 0's. It is unchanged.