# Lesson 4.02 — For-Each Loop & Arrays Class

## Overview

**Objectives** — ***Students will be able to…***

- **Populate and access** arrays using a for-each loop

**Assessments** — ***Students will…***

- **Complete** manipulatives exercises on WS 4.2

**Homework** — ***Students will…***

- **Read** BJP 7.2 up to "Reversing an Array"
- **Complete** self-check questions #12-14

## Materials & Prep

- **Projector and computer**
- **Whiteboard and markers**
- **Classroom copies** of WS 4.2
- **Poster** 4.2
- **Array whiteboards** (see notes at end of LP 4.2) **and dry erase markers**
- **Student small-group assignments** ( 3-4 students per group)
- **Large manipulative** for teacher demo (**optional**)

## Pacing Guide

| Section | Total Time |
|---|---:|
| Bell-work and attendance | 5min |
| Introduction to Arrays | 10min |
| Student Array Activity | 35min |
| Paper selection & grade announcement | 5min |

## Procedure

As with yesterday's class, divide your class into small groups, and have the manipulatives out at each group station, desk, or workspace. Offer a just-in-time intro to for-each loops, but do not get bogged down in the introductory lesson. Many students will learn from physically practicing the for-each control structure using the array whiteboards. Circulate around the room to check for understanding, but let students help and challenge each other.

### Bell-work and Attendance [5 minutes]

### Introduction to Arrays [10 minutes]

1. Briefly introduce the **for-each loop**, **array initialization**, and the **Arrays class** before dismissing the class to work on their group activity:

   If you want to access each element in the array without changing the values (to summarize or count them), you can access them using a for-each loop:

   ```
   for (<type> <name> : <array>){
       <statement>;
       <statement>;
       …
   }
   ```

2. Place an array on the board, demonstrating the quick way to declare an array, and then illustrate how a for-each loop could be used to access the array:

   ```
   int[] fallTemperatures = {55, 50, 59, 69, 48, 30, 48};
   ```

   - This initializes an array called fallTemperatures with 7 integer values. Ask students when they might see these temperatures in their region, and how the values in the array would differ during another season or in a different location. It may make sense in your region to change the array name to winterTemperatures or nightTimeTemperatures, etc.

     ```
     for (int i = 0; i < fallTemperatures.length; i++) {
         if (fallTemperatures[i] > 32) {
             above++;
         }
     }
     ```

   - This is our traditional loop, which traverses the array and sums up all the temperatures that are above freezing (we assume there is a method called "above" that keeps a running count of how many days were above 32 degrees). We can express this same process with a for-each loop:

     ```
     for (int i : fallTemperatures) {
         if (i > 32) {
             above++;
         }
     }
     ```

     with the general form of:

     ```
     for (<type> <name> : <array>) {
         <statement>;
         <statement>;
         …
     }
     ```

   - Make a point of having your students write a note to remind themselves that for-each loops cannot modify values within an array, only examine each value in sequence.

**Student Array Activity [35 minutes]**

1. Distribute white boards and markers or paper printouts to group workstations before students get seated.
2. If you feel that your students need the additional structure, assign groups to work together.
3. If you think students will need the extra guidance, help students together in a whole-group setting. To help students without giving them the answer outright, point out that:
   a. `numbers[7]` evaluates to `0`.
   b. `numbers[numbers[7]]` → `numbers[0]` does NOT evaluate to 0.
   c. Instead, the memory location `numbers[0]` (the index 0 of the numbers array) receives a value.
4. Walk around the room, spot-checking for student understanding and answering any student questions.

**Paper Selection & Grade Announcement [5 minutes]**

At the end of class, consider choosing one group's whiteboard to evaluate as a demonstration for the class.

## Accommodation and Differentiation

If you have a student that would benefit from additional tactile or visiospatial learning exercises, invite him or her to create a classroom poster that contains the same information as Poster 4.2.

Rather than assigning groups randomly or by ability, use tiered grouping as a differentiation strategy. Concepts/skills will be covered at different levels of complexity in response to diagnosed needs of each learner. Your tiered group assignments will probably end up changing from one exercise to another, since students' needs and strengths vary with instructional objectives and task types.

## Additional Resources

- CS Awesome Array Algorithms https://runestone.academy/runestone/books/published/csawesome/Unit6-Arrays/topic-6-4-array-algorithms.html

## Video

- CSE 142, *Array Traversal* (26:06–33:26) https://www.youtube.com/watch?v=6M3KEfFpYiM&start=1570
- CSE 142, *For-Each Loop* (19:40–22:40) https://www.youtube.com/watch?v=6M3KEfFpYiM&start=1180

## Forum discussion

Lesson 4.02 For-Each Loop & Arrays Class (TEALS Discourse account required)