

Lesson 4.01 — Array Basics

Overview

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

- Define, populate, and access arrays.

Assessments — *Students will...*

- **Complete** exercises with manipulatives on WS 4.1.

Homework — *Students will...*

- **Read** BJP 7.1 “For-Each Loop” and “The Arrays Class”
- **Complete** self-check questions #1, 7, 9

Materials & Prep

- **Projector and computer**
- **Whiteboard and markers**
- **Classroom copies** of WS 4.1, Poster 4.2
- **Array whiteboards** (see notes at end of lesson) **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	20min
Student array activity	25min
Paper selection & grade announcement	5min

Procedure

Divide your class into small groups today, and have the manipulatives (see below) out at each group station, desk, or workspace. Offer a just-in-time intro to arrays as outlined below, but don’t belabor instruction; students may have better success physically working through the activity. 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 [20 minutes]

Emphasize with students...

Content - Uses of pre-built data structures An array is a data structure that is available for a programmer to use in Java. You will find that there are situations where arrays will work perfectly for your program and they will facilitate finding a solution. Other times an array might not be the best data structure to use and you might have to use something different.

As you complete this activity, think carefully about how arrays could be used in your future programs.

1. Open up with the example of daily temperature on the slides.

- Ask the class to build the program, they should reach the point where they calculate the average, but are unable to calculate the days above average as they must access the data a second time.
- In order to successfully make the program, we would need to store the temperature for every day.
- An array is an indexed structure that holds multiple values of the same type. Ask students if they've seen anything in Java that might be an array. (A String can be thought of as an array of characters!)
- The values stored in an array are called elements. Individual elements are accessed using an integer index (the position). Ask students what element is stored at index 2, 4, and 7 in this string/array.
- Since an array is an object, you have to construct it (you can't just declare it as a variable).

```
int[] numbers = new int[10];
```

Nothing is in the highlighted brackets because you're describing what type is going to be contained in the array.

```
int[] numbers = new int[10];
```

Here's the name of your array—in this case we're making an array of numbers.

```
int[] numbers = new int[10];
```

__We use the `* new *` keyword since we're constructing an object, then we tell Java how many elements we want to store in our array. In this case we want to store 10 numbers in the array.__

- Check for student understanding by asking students to tell you how to construct an array that holds 9 integers.

```
int[] numbers = new int[9];
```

- Alternatively, you can initialize an array by writing out the full array.

```
int[] numbers = {0, 1, 2, 3, 4, 5};
```

- Both of these sample arrays are only ½ done right now—they're arrays filled with 0s because Java auto-initializes arrays to a default value of 0 (for `char`, `double`, and `int`) or `false` (for `boolean`). So our *number* array looks something like this:

```

.------.---.---.---.
| array `numbers`: | 0 | 0 | 0 | 0 |
+-----+-----+-----+
|      index      | 0 | 1 | 2 | 3 |
'-----'-----'-----'

```

- To fill in this array, we need to fill in the values for each location:

```
numbers[0] = 27;
numbers[3] = -6;
```

- Now the array looks like this:

```

.------.---.---.---.
| array `numbers`: | 27| 0 | 0 |-6 |
+-----+-----+-----+
|      index      | 0 | 1 | 2 | 3 |
'-----'-----'-----'

```

- Now that students can create an array of integers, ask the students how would they create an array of temperatures and calculate the average temperature? What intermediate step would be needed before calculating the average temperature?

- Briefly touch on other types of arrays and common errors.

- You can have arrays of almost anything: `String`, `double`, `boolean`, *etc.* Examples of an instantiated double and boolean array are on the slides. Ask the class what they must change to create these arrays.

- Cover the common index-out-of-bounds exception. If the program calls illegal indexes or indexes outside 0 and the array's length-1, Java will throw you an exception. (It's always nice to go over reasons for exceptions so you don't need to correct them all later!)
4. If you have a really big array, you can use a Scanner to grab values from user input, or you can autofill them with a loop:

```
for (int i = 0; i < age.length; i++){
    age[i] = input.nextInt();
}
```

As you move across the array (in this case to fill each element with a user-inputted value), we call this “array traversal.” You'll need to do this a lot in the future, so you should put a general formula in your Tricky Code Cheat Sheet:

```
for (int i = 0; i < age.length; i++){
    // do something with age[i];
}
```

Student Array Activity [25 minutes]

1. Distribute white boards and markers to group workstations before students get seated.
2. If you feel that your students need the additional structure, assign groups to work together on Problem 2.
3. 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

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.

You can reduce paper waste and increase student engagement by creating an inexpensive classroom set of “array whiteboards” following these instructions:

1. Purchase panel board from your local hardware store (<http://tinyurl.com/zgtlbhr>)
2. Have the assistant cut the board into long strips that you can use as 1 dimensional arrays.
 1. If you like to use individual whiteboards to check for student understanding, you can have whiteboards cut to individual student squares, then have students line up the little white boards into a one-dimensional array for this exercise.
 2. If you plan on using these as array boards, you can either subdivide the boards into element-blocks with black electrical tape, or you can have your students draw the blocks in with their dry erase markers. (The latter option encourages students to construct arrays of different sizes.)
 3. Your total number of array white boards should be:

$$(\# \text{ small groups in your classroom}) * 2 + 1 \text{ instructor array board}$$
3. Use these array-whiteboards to demonstrate the relationship between 1 (and later 2) dimensional arrays during this unit.

In mathematics, a **manipulative** is an object which is designed so that a learner can perceive some mathematical concept by manipulating it, hence its name. The use of manipulatives provides a way for children to learn concepts in a developmentally appropriate, hands-on and experiential way.

[TEST: hello world]

Teacher Prior CS Knowledge

Arrays in Java (and other object oriented programming languages) are classes. This brings the whole object oriented paradigm into play. When declaring an array variable, the variable is now a reference to an array object. In order to create an object from a class the programmer uses the **new** keyword. There is a distinction between the array reference and the array object. This programming construct allows for multiple reference to point to the same object.

Common Mistakes

Arrays common mistakes: <http://interactivepython.org/runestone/static/JavaReview/ArrayBasics/aMistakes.html>

Misconceptions

- Students understanding of the difference between the index `i` and the content of the `i`th element stored in `a[i]`.
- Loop bounds:
 - 0-based index arrays,
 - where arrays end, versus array length, and
 - what are the range of indices to express a particular array range.

Video

- BJP 7-1, *Array Simulation* http://media.pearsoncmg.com/aw/aw_reges_bjp_2/videoPlayer.php?id=c7-1
- CSE 142, *Arrays* (1:35-26:06) <https://www.youtube.com/watch?v=6M3KEffpYiM&start=95>
- CS Homework Bytes, *Arrays, with Ariel* <https://www.youtube.com/watch?v=PFohS2HvCgs>

Forum discussion

Lesson 4.01 Array Basics (TEALS Discourse account required)