

Lesson 4.03 — Printing, Searching, & Testing for Equality

Overview

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

- **Manipulate** single-dimension arrays using a variety of array transversal algorithms.

Assessments — *Students will...*

- **Teach** a mini-lesson on printing, searching/replacing, testing for equality, reversing an array, or string traversal.
- **Complete** a quiz at the end of Day 2

Homework — *Students will...*

- **Day 1:** Complete self-check questions #15-17 and exercise 3
- **Day 2:** Read BJP 7.3 and complete self-check questions #19-21

Materials & Prep

- **Group copies** of WS 4.3
- **Assignments** for 5 student groups
- **5 classroom copies** of the textbook (or have students bring their copies to class)
- **Copies of the grading rubric** (on the overhead or printed out; **optional**)

You will need to circle student assignments on point 2 of WS 4.3, so each group knows what topic they are expected to teach.

Pacing Guide: Day 1

Section	Total Time
Bell-work and attendance	5min
Introduction to assignment	10min
Student preparation of lesson & quiz questions	40min

Pacing Guide: Day 2

Section	Total Time
Group set-up, attendance	5min
Group presentations (~8 minutes per group)	40min
Quiz	10min

Procedure

Your hook for today's lesson is to turn the reigns over to students immediately. Have instructions printed out and sitting at teamwork stations (or on student desks). Encourage students to answer their own questions using the instruction sheet and textbook. Frequently asked questions and suggestions for student groups are included in Accommodations and Differentiation.

Bell-work and Attendance [5 minutes]

Introduction to Assignment [10 minutes]

1. Review the assignment as outlined on WS 4.3, and ask students to explain back to you what their group presentations need to include for full credit. If you need to, review procedures for group work, then split the

class into groups.

If you feel it will benefit your class, reviewing or distributing the grading rubric before the assignment should be done at this point.

Student Preparation of Lesson and Quiz Questions [40 minutes]

Give students the class period to prepare their presentation and generate quiz questions for tomorrow's class.

- Use a timer and periodically announce how much time is left in class so students can pace themselves.
- Emphasize to students that they should primarily draw upon their textbooks to help them plan their lesson.
- Answers to commonly asked questions, and tips for the different groups may be found below:

FAQ: PRINTING AN ARRAY

Guidance for student teachers

1. If students are lacking direction, encourage students to refer to the textbook and trace the loop and predict the output of the code that prints the array "list" vertically.
2. A good "Tricky Code Cheat Sheet" tip can be found on the same page; any time you see a list of values separated by commas, students should remember to use a fencepost algorithm.

Common Questions/Answers

1. **When would we encounter an empty array?**

If you set array values to null, they're actually empty (not just equal to 0 or false). Another way you could get an empty array is if you initialized an array of size 0; then there would be no elements at all!

2. **Why do we have to include extra code to account for an empty array?**

We put in "contingency code" to deal with an empty array to be thorough. If we don't handle special cases correctly, Java will throw an exception, so we try to cover our bases and prevent that from happening.

FAQ: SEARCHING AND REPLACING

Guidance for student teachers

1. If students are having trouble outlining their lesson, encourage them to teach counting, then locating values in a list. Once they've covered those methods, they can finish up by tracing the sample method replaceAll and explaining the output.
2. A good "Tricky Code Cheat Sheet" tip would be the convention of returning -1 if a value is not found in the list.

Common Questions/Answers

1. **Why do we use a for-each loop for counting occurrences, but not for finding values in a list?**

*Neither one requires you to change the contents of the array! **If we're finding an item in a list and returning its index, then we need to track indexes as we search. That's done naturally with a regular for loop; with a for-each loop, we'd have to manually simulate a for loop to compute the index.*

2. **Why is the method "count" initialized to zero?**

This is setting up the loop to work correctly—review "priming the pump" analogy if needed.

FAQ: TESTING FOR EQUALITY

Guidance for student teachers

1. Remind students to include the definition for what makes arrays equivalent (they have the same length and store the same sequence of values).
2. If students are struggling with the sequence of their lesson, you might suggest that they begin by teaching the sample method "equals," then circling back to explain why testing for inequality is easier than testing for equality.
3. A good general tip for the Tricky Code Cheat Sheet would be the common pattern for methods like "equals;" Test all for the ways that the two objects might not be equal, returning false if there are any differences, or true at the very end if all tests are passed.

Common Questions/Answers

1. **Can't we just use Arrays.equals to see if two arrays are equal?**

Yes, you can (and should) use this method from the Arrays class. In teaching your unit, your purpose is to familiarize students with the way to write code that tests equals in general. While you might not need it in this particular situation, you may want to tweak the method down the road, and you cannot edit the Arrays.equals method because it's not in a class you can edit.

FAQ: REVERSING AN ARRAY

Guidance for student teachers

1. If students need guidance on structuring their lesson, encourage them to work through the final (correct) method “swap,” trace the flow of control and output, then cover why other versions don’t work. (They may not present in this order, but covering the correct answer should help them organize their thoughts.)
2. A good tip for the Tricky Code Cheat Sheet is to summarize the steps needed to swap two values. This comes up often on the AP exam.

Common Questions/Answers

1. Why can’t we just swap values by assigning them to each other?

If you assign one value to the other, you copy over the values, so ½ way through the process, you’ve got 2 copies of the same value. The second half of the swap doesn’t work!

2. Why do we have to stop the swap loop ½ way through the list.length? Won’t that just swap ½ of the list?

Using a line of 4 – 6 objects (pens, paperclips, whatever you have around the classroom), start by swapping the first & last items, then the next-inner-two, and so on. Count each move as you make it, then pause at the halfway point, pointing out you’ve conducted length/2 swaps. Proceed with the final swaps, returning objects back to their original place.

FAQ: STRING TRAVERSAL ALGORITHMS

Guidance for student teachers

1. Encourage the group to spend some time during their lesson on why you use parentheses for most arrays, but not with strings.
2. If students are at a loss for how to teach their segment, suggest covering an example and a non-example for contrast. A good non-example would be traversing an array that isn’t a string.

Common Questions/Answers

None.

1. Collect quiz questions before the end of class. Check and compile into a quiz for the end of Day

2.

2. On Day 2, give each group 5 minutes to present their topic and 3 minutes for questions.

Encourage students to ask questions, and be sure to ask a question or two of each team (depending on how many teams you have).

3. Use the grading rubric as outlined here:

3 pts.	2 pts.	1 pts.	0 pts.
Presentation includes definitions and an example with proper syntax.	Presentation includes definitions or an example with proper syntax.	Presentation includes definitions or an example with proper syntax with few mistakes.	Presentation includes definitions or an example with proper syntax with many mistakes.
Presentation includes a non-example as helpful contrast.	Presentation includes a non-example that is marginally helpful.	Presentation includes a non-example that does not add to comprehension.	Presentation includes a non-example that adds confusion, or presentation does not include a non-example.
Presentation includes a helpful tip that is clearly explained and concisely stated.	Presentation includes a helpful tip that is clearly explained or concisely stated.	Presentation includes a helpful tip that is not clearly explained and may include a small error.	Presentation does not include a helpful tip or hint.

4. Administer the student-generated quiz to assess student understanding.

Accommodation and Differentiation

Circle around the room to help students through reading the text in the textbook. Make sure that each of your working teams are properly stratified (rather than using tiered grouping).

If students are speeding along, encourage students to write down questions to pose to other groups during mini-lessons. If everyone finishes creating their lessons early, start the classroom presentations on Day 1 instead of waiting for Day 2. If only 1 or 2 groups have finished early, encourage groups to rehearse lesson delivery.

Video

- BJP 7-2, *Array Traversal Algorithms* http://media.pearsoncmg.com/aw/aw_reges_bjp_2/videoPlayer.php?id=c7-2
- CSE 142, *Random Access to Array* (33:26-42:00) <https://www.youtube.com/watch?v=6M3KEffpYiM&start=2006>

Forum discussion

Lesson 4.03 Printing, Searching, & Testing for Equality (TEALS Discourse account required)