21.7 Loop-modifying or copying/comparing arrays

This section has been set as optional by your instructor.

Modifying array elements

A program may need to modify elements while iterating through an array. The program below uses a loop to convert any negative array element value to 0.

Figure 21.7.1: Modifying an array during iteration example: Converting negatives to 0 program.

```
import java.util.Scanner;

public class NegativeToZero {
   public static void main(String[] args) {
      Scanner scnr = new Scanner(System.in);
      final int NUM_ELEMENTS = 8;              //
Number of elements
      int[] userVals = new int[NUM_ELEMENTS]; //
User numbers
      int i;                                   //
Loop index

      // Prompt user to input values
      System.out.println("Enter " + NUM_ELEMENTS + "
integer values...");
      for (i = 0; i < userVals.length; ++i) {
         System.out.print("Value: ");
         userVals[i] = scnr.nextInt();
      }

      // Convert negatives to 0
      for (i = 0; i < userVals.length; ++i) {
         if (userVals[i] < 0) {
            userVals[i] = 0;
         }
      }

      // Print numbers
      System.out.print("New numbers: ");
      for (i = 0; i < userVals.length; ++i) {
         System.out.print(userVals[i] + " ");
      }
   }
}
```

```
Enter 8 integer
values...
Value: 5
Value: 67
Value: -5
Value: -4
Value: 5
Value: 6
Value: 6
Value: 4
New numbers: 5 67 0
0 5 6 6 4
```

Feedback?

PARTICIPATION ACTIVITY 21.7.1: Modifying an array in a loop.

What is the resulting array contents, assuming each question starts with an array of size 4 having contents -55, -1, 0, 9?

1)
```
for (i = 0; i < 4;
++i) {
   itemsList[i] = i;
}
```

- ○ -54, 0, 1, 10
- ◉ 0, 1, 2, 3
- ○ 1, 2, 3, 4

Correct

First iteration assigns 0 to element 0. Next assigns 1 to element 1. Etc.

2)

```
for (i = 0; i < 4; ++i) {
   if (itemsList[i] < 0) {
      itemsList[i] = itemsList[i] * -1;
   }
}
```

- -55, -1, 0, -9
- 55, 1, 0, -9
- 55, 1, 0, 9

Correct

Negative elements are negated, making them positive.

3)

```
for (i = 0; i < 4; ++i) {
   itemsList[i] = itemsList[i+1];
}
```

- -1, 0, 9, 0
- 0, -55, -1, 0
- Out-of-range access

Correct

When i is 3, the loop tries to assign element 3+1, or 4, to element 3. But element 4 does not exist, yielding an out-of-range access.

4)

```
for (i = 0; i < 3; ++i) {
   itemsList[i] = itemsList[i+1];
}
```

- -1, 0, 9, 9
- Out-of-range access
- -1, 0, 9, 0

Correct

For each iteration, the next element is assigned to the current element. The loop expression exits before the last element.

5)

```
for (i = 0; i < 3; ++i) {
   itemsList[i+1] = itemsList[i];
}
```

- -55, -55, -55, -55
- -55, -55, -1, 0
- Out-of-range access

Correct

After iteration with i = 0: -55, -55, 0, 9. With i = 1: -55, -55, -55, 9. Etc. New programmers commonly forget that each iteration deals with the latest modified array.

Feedback?

zyDE 21.7.1: Modifying an array during iteration example: Doubling element values.

Complete the following program to double each number in the array.

Load default template...

```
11          System.out.println
12          for (i = 0; i < us
13             System.out.prin
14             userVals[i] = 
15          }
16
17          // Double each el
18
19          // Print numbers
20          System.out.print(
21          for (i = 0; i < us
22             System.out.prin
23          }
24       }
25    }
26
```

5 67 -5 -4 5 6 6 4

Run

Feedback?

Copying an array

Copying an array is a common task. Given a second array of the same size, a loop can copy each element one-by-one. Modifications to either array do not affect the other.

Figure 21.7.2: Array copying: Converting negatives to 0 program.

```
import java.util.Scanner;

public class NegativeToZeroCopy {
   public static void main(String[] args) {
      Scanner scnr = new Scanner(System.in);
      final int NUM_ELEMENTS = 8;                //
Number of elements
      int[] userVals = new int[NUM_ELEMENTS];    //
User numbers
      int[] copiedVals = new int[NUM_ELEMENTS]; // New
numbers
      int i;                                     //
Loop index

      // Prompt user for input values
      System.out.println("Enter " + NUM_ELEMENTS + "
integer values...");
      for (i = 0; i < userVals.length; ++i) {
         System.out.print("Value: ");
         userVals[i] = scnr.nextInt();
      }

      // Convert nums to newNums
      for (i = 0; i < userVals.length; ++i) {
         copiedVals[i] = userVals[i];
      }

      // Convert negatives to 0
      for (i = 0; i < copiedVals.length; ++i) {
         if (copiedVals[i] < 0) {
            copiedVals[i] = 0;
         }
      }

      // Print numbers
      System.out.println("\nOriginal and new values:
");
      for (i = 0; i < userVals.length; ++i) {
         System.out.println(userVals[i] + " became " +
copiedVals[i]);
      }
      System.out.println();
   }
}
```

```
Enter 8 integer
values...
Value: 12
Value: -5
Value: 34
Value: 75
Value: -14
Value: 33
Value: 12
Value: -104

Original and new
values:
12 became 12
-5 became 0
34 became 34
75 became 75
-14 became 0
33 became 33
12 became 12
-104 became 0
```

Feedback?

PARTICIPATION ACTIVITY 21.7.2: Array copying.

Given array firstList with size 4 and element values, 33, 44, 55, 66, and array secondList with size 4 and elements values 0, 0, 0, 0.

1) firstList = secondList copies 0s into each firstList element.

- ○ True
- ◉ False

Correct

Array copying doesn't work that way. A loop is needed to copy each element one-by-one.

2) This loop copies firstList to secondList, so that secondList becomes 33, 44, 55, 66:

```
for (i = 0; i < 4;
++i) {
   secondList[i] =
firstList[i];
}
```

- ◉ True
- ○ False

Correct

Element 0 is copied. Then element 1. Then 2. Then 3.

3) After a for loop copies firstList to secondList, the assignment secondList[0] = 99 will modify both arrays.

- ○ True
- ◉ False

Correct

secondList is an independent copy. Changes to either array do not affect the other.

4) Given thirdList with size 5 and elements 22, 21, 20, 19, 18, the following causes firstList's values to be 22, 21, 20, 19, 18:

```
for (i = 0; i < 5;
++i) {
   firstList[i] =
thirdList[i];
}
```

- ○ True
- ◉ False

Correct

The last iteration will attempt firstList[4] = thirdList[4]. But element firstList[4] doesn't exist (the array's size is 4, so last element is firstList[3]), an error occurs. Copying arrays should involve same sized array (or a smaller array copied to the beginning of a larger array).

Feedback?

Write a loop that subtracts 1 from each element in lowerScores. If the element was already 0 or negative, assign 0 to the element. Ex: lowerScores = {5, 0, 2, -3} becomes {4, 0, 1, 0}.

Learn how our autograder works

437612.2739130.qx3zqy7

```
5       final int SCORES_SIZE = 4;
6       int[] lowerScores = new int[SCORES_SIZE];
7       int i;
8
9       for (i = 0; i < lowerScores.length; ++i) {
10         lowerScores[i] = scnr.nextInt();
11      }
12
13      /* Your solution goes here  */
14
15      for (i = 0; i < lowerScores.length; ++i) {
16         System.out.print(lowerScores[i] + " ");
17      }
18      System.out.println();
19   }
20 }
```

Run

Feedback?

```
1 import java.util.Scanner;
2 public class StudentScores {
3    public static void main (String [] args) {
```

```
4       Scanner scnr = new Scanner(System.in);
5       final int SCORES_SIZE = 4;
6       int[] oldScores = new int[SCORES_SIZE];
7       int[] newScores = new int[SCORES_SIZE];
8       int i;
9
10      for (i = 0; i < oldScores.length; ++i) {
11         oldScores[i] = scnr.nextInt();
12      }
13
14      /* Your solution goes here  */
15
16      for (i = 0; i < newScores.length; ++i) {
```

Run

Feedback?

```
5       final int SCORES_SIZE = 4;
6       int[] bonusScores = new int[SCORES_SIZE];
7       int i;
8
9       for (i = 0; i < bonusScores.length; ++i) {
10         bonusScores[i] = scnr.nextInt();
11      }
12
13      /* Your solution goes here  */
14
15      for (i = 0; i < bonusScores.length; ++i) {
16         System.out.print(bonusScores[i] + " ");
17      }
18      System.out.println();
19   }
20 }
```

Run

Feedback?

CHALLENGE ACTIVITY | 21.7.4: Modify an array's elements.

Double any element's value that is less than controlValue. Ex: If controlValue = 10, then dataPoints = {2, 12, 9, 20} becomes {4, 12, 18, 20}.

Learn how our autograder works

437612.2739130.qx3zqy7

```
8       int i;
9
10      controlValue = scnr.nextInt();
11
12      for (i = 0; i < dataPoints.length; ++i) {
13         dataPoints[i] = scnr.nextInt();
14      }
15
16      /* Your solution goes here  */
17
18      for (i = 0; i < dataPoints.length; ++i) {
19         System.out.print(dataPoints[i] + " ");
20      }
21      System.out.println();
22   }
23 }
```

Run

Feedback?

How was this section? Provide section feedback