



Computer Science Virtual Learning

HS Computer Science A

May 13th, 2020



Lesson: Using a For Loop to Loop Through an Array

Objective/Learning Target:

Understanding how to use a for loop to loop through all the elements of an array in Java.

Understanding how to loop from Back to Front through an array in Java.

For Loops in Arrays

You can also use a for loop to loop through all the elements of an array. Just start the index at 0 and loop while the index is less than the length of the array.

A screenshot of an IDE window with a yellow background. At the top right, there is a "Run" button and a slider set to "Original - 1 of 1". The code editor contains two methods: `multAll` and `printValues`. `multAll` uses a for loop to iterate over an array and multiply each element by a given amount. `printValues` uses a for loop to iterate over an array and print each element. Below the code editor is a console window showing the output of the `printValues` method: 4, 12, 14, 24, 10.

```
9
10 public void multAll(int amt)
11 {
12     for (int i = 0; i < values.length; i++)
13     {
14         values[i] = values[i] * amt;
15     } // end for loop
16 } // end method
17
18 public void printValues()
19 {
20     for (int val : values )
21     {
22         System.out.println(val);
23     }

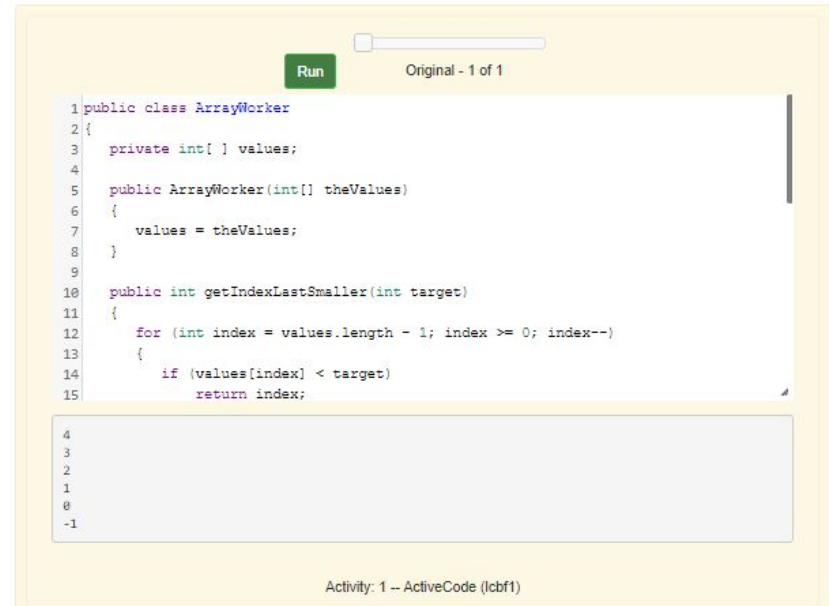
```

```
4
12
14
24
10
```

Activity: 1 -- ActiveCode (lcal1)

Looping From Back to Front

You don't have to loop through an array from the front to the back. You can loop by starting at the back of the array and move toward the front during each time through the loop. This can be handy when you are looping through a sorted array and want to find the index of the last number that is less than some given number as shown in the method `getIndexLastSmaller` below. Notice that the method returns `-1` if there is no number in the array that is smaller than the given number. Why does this work?

A screenshot of an IDE window titled "Original - 1 of 1" with a "Run" button. The code defines a class `ArrayWorker` with a `private int[] values;` field. The `ArrayWorker(int[] theValues)` constructor assigns `values = theValues;`. The `getIndexLastSmaller(int target)` method uses a `for` loop starting at `values.length - 1` and decrementing `index--` until it finds a value less than the target, returning that index, or `-1` if none is found. The output console shows the values `4, 3, 2, 1, 0, -1`.

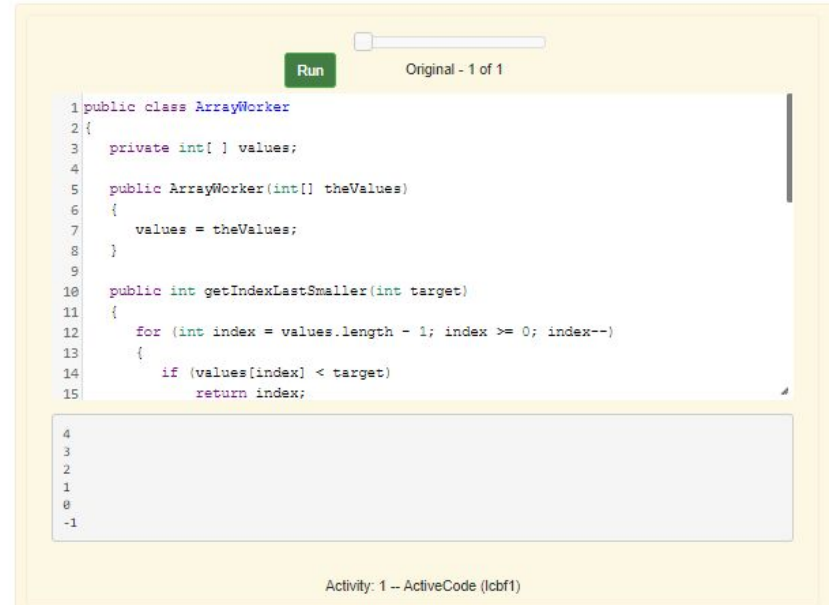
```
1 public class ArrayWorker
2 {
3     private int[] values;
4
5     public ArrayWorker(int[] theValues)
6     {
7         values = theValues;
8     }
9
10    public int getIndexLastSmaller(int target)
11    {
12        for (int index = values.length - 1; index >= 0; index--)
13        {
14            if (values[index] < target)
15                return index;
16        }
17    }
18 }
```

4
3
2
1
0
-1

Activity: 1 -- ActiveCode (lcbf1)

Looping from Back to Front

Notice that if the array is a field of the ArrayWorker class you must create an ArrayWorker object in the main method. You don't have to pass the array to the `getIndexLastSmaller` method like you do if the method is static. The object already has the array as a field and any object method has access to it.

A screenshot of a code editor window. At the top right, there is a green "Run" button and a progress indicator showing "Original - 1 of 1". The code is as follows:

```
1 public class ArrayWorker
2 {
3     private int[] values;
4
5     public ArrayWorker(int[] theValues)
6     {
7         values = theValues;
8     }
9
10    public int getIndexLastSmaller(int target)
11    {
12        for (int index = values.length - 1; index >= 0; index--)
13        {
14            if (values[index] < target)
15                return index;
```

The bottom part of the editor shows the output of the program, which is a list of integers: 4, 3, 2, 1, 0, -1.



Check Your Understanding

1. Given the following code segment what will be returned when you execute: `getIndexLastSmaller(-13)`;

```
private int[] values = {-20, -15, 2, 8, 16, 33};
public int getIndexLastSmaller(int compare)
{
    for (int i = values.length - 1; i >=0; i--)
    {
        if (values[i] < compare) return i;
    }
    return -1; // to show none found
}
```

- A. -1
- B. -15
- C. 1
- D. You will get an out of bounds error.

2. Given the following code segment what will be returned when you execute: `getIndexLastSmaller(7)`;

```
private int[] values = {-20, -15, 2, 8, 16, 33};

public int getIndexLastSmaller(int compare)
{
    for (int i = values.length; i >=0; i--)
    {
        if (values[i] < compare) return i;
    }
    return -1; // to show none found
}
```

- A. -1
- B. 1
- C. 2
- D. You will get an out of bounds error.



For More Resources and to Check Answers

Go to: <https://runestone.academy/runestone/books/published/apcsareview/ArrayBasics/aLoopBackToFront.html>