



Computer Science Virtual Learning

HS Computer Science A

May 14th, 2020



Lesson: Looping Through Part of an Array

Objective/Learning Target:

Understanding how to loop through part of an array in java
Recognizing common mistakes when using arrays in java

Looping Through Part of an Array

You don't have to loop through all of the elements of an array. You can loop through just some of the elements of an array using a for loop. The following code doubles the first five elements in an array. Notice that it uses a complex conditional (`&&`) on line 12 to make sure that the loop doesn't go beyond the bounds of the array.

A screenshot of a code editor window. At the top right, there is a "Run" button and a progress indicator labeled "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 void doubleFirstFive()
11    {
12        for (int i = 0; i < values.length && i < 5; i++)
13        {
14            values[i] = values[i] * 2;
15        }
16    }
17 }
```

Below the code editor, there is a console output area showing the following values:

```
6
16
-6
4
40
5
33
1
```

At the bottom of the editor window, it says "Activity: 1 -- ActiveCode (lclp1)".

Practice: Looping Through Part of an Array

You can even start in the middle and loop through the rest of the array. Does this work for arrays that have an even number of elements? Does it work for arrays that have an odd number of elements?

Go to

<https://runestone.academy/runestone/books/published/apcsareview/ArrayBasics/aLoopPart.html>

Modify the second set of code to test with both arrays with an even number of items and an odd number.

A screenshot of a Java code editor interface. At the top right, there is a "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 void doubleLastHalf()
11    {
12        for (int i = values.length / 2; i < values.length; i++)
13        {
14            values[i] = values[i] + 2;
15        }
16    }
17 }
```

Below the code editor is a console window with the following output:

```
3
8
-6
4
```

At the bottom of the editor, it says "Activity: 2 -- ActiveCode (lclp2)".



Things to Watch for When Looping Through an Array

When processing all array elements be careful to start at the first index which is 0 and end at the last index which is `arrayName.length - 1`. Be careful not to go past the bounds of the array which means don't use a negative number as an index or a number that is equal to or greater than the length of the array.

Practice: Things to Watch For When Looping Through an Array

Also, be careful not to jump out of loop too early when you are looking for a value in an array. The method below uses **return** statements to stop the execution of the method and return a value to the method that called this method. If a return statement returns a value, the type of that value must match the return type in the method header. Methods with a return type of **void** can't return any values, but can have one or more return statements.

What is wrong with this code? The first time through the loop it will start with the element at index 0 and check if the item at the array index equals the passed target string. If they have the same characters in the same order it will return 0, otherwise it will return -1. But, it has only processed one element of the array. How would you fix the code to work correctly (process all array elements before returning)?

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

```
1 public class StringWorker
2 {
3     private String[] arr = {"Hello", "Hey", "Good morning!"};
4
5     public int findString(String target)
6     {
7         String word = null;
8         for (int index = 0; index < arr.length; index++)
9         {
10            word = arr[index];
11
12            if (word.equals(target))
13            {
14                return index;
15            }
16        }
17    }
18 }
```

Below the code editor, there is a text input field containing the value "-1". At the bottom right of the editor, it says "Activity: 1 -- ActiveCode (Icap1)".



Common Mistakes

- forgetting to create the array - only declaring it (`int[] nums;`)
- using 1 as the first index not 0
- using `array.length` as the last valid index in an array, not `array.length - 1`.
- using `array.length()` instead of `array.length` (not penalized on the free response)
- using `array.get(0)` instead of `array[0]` (not penalized on the free response)
- going out of bounds when looping through an array (using `index <= array.length`). You will get an `ArrayIndexOutOfBoundsException`.
- jumping out an loop too early by using one or more return statements before every value has been processed.



Check Your Understanding

1. Given the following values of `a` and the method `doubleLast` what will the values of `a` be after you execute: `doubleLast()`?

```
private int[ ] a = {-20, -15, 2, 8, 16, 33};
public void doubleLast()
{
    for (int i = a.length / 2; i < a.length; i++)
    {
        a[i] = a[i] * 2;
    }
}
```

- A. {-40, -30, 4, 16, 32, 66}
- B. {-40, -30, 4, 8, 16, 32}
- C. {-20, -15, 2, 16, 32, 66}
- D. {-20, -15, 2, 8, 16, 33}

2. Given the following values of `a` and the method `mystery` what will the values of `a` be after you execute: `mystery()`?

```
private int[ ] a = {-20, -15, 2, 8, 16, 33};
public void mystery()
{
    for (int i = 0; i < a.length/2; i+=2)
    {
        a[i] = a[i] * 2;
    }
}
```

- A. {-40, -30, 4, 16, 32, 66}
- B. {-40, -30, 4, 8, 16, 33}
- C. {-20, -15, 2, 16, 32, 66}
- D. {-40, -15, 4, 8, 16, 33}
- E. {-40, -15, 4, 8, 32, 33}



For More Resources and to Check Answers

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