



**Computer Science Virtual Learning**

# **HS Computer Science A**

**May 11th, 2020**

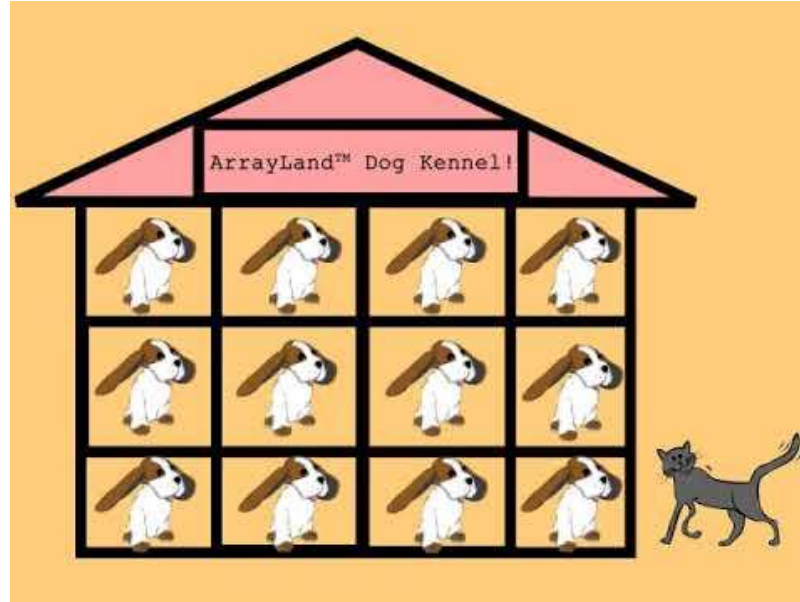


Lesson: **What is an Array?**

**Objective/Learning Target:**

Understanding what an Array is and the how to apply it using  
Java

# What is an Array?



## What is an Array?

An array is consecutive storage for multiple items of the same type. You can store a value in an array using an index (location in the array). You can get a value from an array using an index. An array is like a row of lockers, except that you can't cram lots of stuff into it. You can only store one value at each array index. An array index is like a locker number. It helps you find a particular place to store your stuff and retrieve stuff.





# What is an Array?

Arrays are useful whenever you have several elements of data of the same type that you want to keep track of, but you don't need to name each one. If you want to keep track of the top 5 highest scores in a game and the names of the people with those scores, you could use two arrays. One array could keep track of the scores and the other the names.

Can you think of another example of something that is like an array?

## Declaring an Array

To declare an array specify the type of elements that will be stored in the array, then ([ ]) to show that it is an array of that type, then at least one space, and then a name for the array. Note that the declarations below just name the variable and say what type of array it will reference. The declarations do not create the array. Arrays are objects in Java, so any variable that declares an array holds a reference to an object. If the array hasn't been created yet and you try to print the value of the variable, it will print null (meaning it doesn't reference any object yet).

A screenshot of a Java IDE window. At the top right, there is a "Run" button and a progress indicator showing "Original - 1 of 1". The main area contains a code editor with the following Java code:

```
1 public class Test1
2 {
3     public static void main(String[] args)
4     {
5         // declare the arrays
6         int[] highScores = null;
7         String[] names = null;
8
9         System.out.println(highScores);
10        System.out.println(names);
11    }
12 }
13
```

Below the code editor is an output window showing the result of running the code:

```
null
null
```

At the bottom of the IDE window, it says "Activity: 3 - ActiveCode (Icab1)".

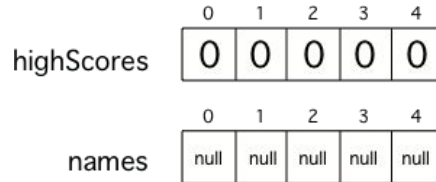


# Creating an Array

To create an array use the new keyword, followed by a space, then the type, and then in square brackets the size of the array (the number of elements it can hold).

```
highScores = new int[5];  
names = new String[5];
```

Array elements are initialized to 0 if they are a numeric type (int or double), false if they are of type boolean, or null if they are an object type like String.



The first value in an array is stored at index 0 and the index of the last value is the length of the array minus one (since the first index is 0).

# Putting Values in an Array

To put a value in an array you give the name of the array and the index number in brackets and then an = and finally the value and a semicolon (`highScores[0] = 99;`). The first item in an array is at index 0.

A screenshot of a Java IDE. At the top right, there is a green "Run" button and a progress indicator labeled "Original - 1 of 1". The main area contains a code editor with the following Java code:

```
1 public class Test1
2 {
3     public static void main(String[] args)
4     {
5         // declare arrays
6         int[] highScores = null;
7         String[] names = null;
8
9         // create the arrays
10        highScores = new int[5];
11        names = new String[5];
12
13        // print the initial values at index 0
14        System.out.println(highScores[0]);
15        System.out.println(names[0]);

```

Below the code editor is a text area showing the output of the program:

```
0
null
99
Jamal
```

At the bottom of the IDE window, it says "Activity: 4 -- ActiveCode (array-set)".

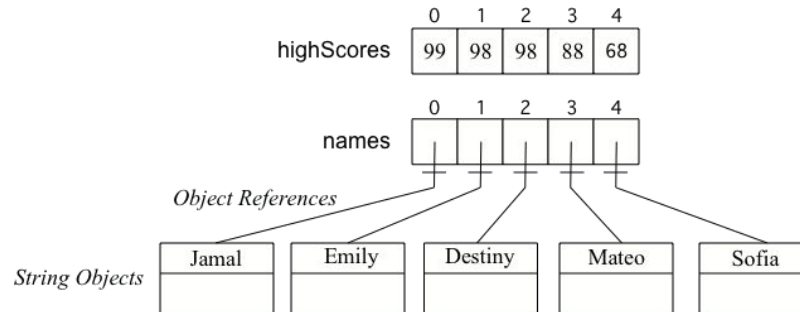


# Initializing Array Values

You can also initialize (set) the values in the array when you create it. In this case you don't specify the size of the array, it will be determined from the number of values that you specify.

```
int[] highScores = {99,98,98,88,68};
String[] names = {"Jamal", "Emily", "Destiny", "Mateo", "Sofia"};
```

When you create an array of a primitive type (like int) with initial values specified, space is allocated for the specified number of items of that type and the values in the array are set to the specified values. When you create an array of an object type (like String) with initial values, space is set aside for that number of object references. The objects are created and the object references set so that the objects can be found.



# Array Length

Arrays know their length (how many elements they can store). It is a public read-only field so you can use dot-notation to access the field (arrayName.length). Dot-notation is using variable name followed by a . and then the field (property) name or a method name.

Note that length is a field and not a method, unlike the String length() method, so you don't add parentheses after length. However, if you use parentheses after length during the exam, you won't lose any points.

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

```
1 public class Test2
2 {
3     public static void main(String[] args)
4     {
5         int[] highScores = {99,99,98,88,68};
6         System.out.println(highScores.length);
7     }
8 }
9
```

Below the code editor is a text area containing the output "5". At the bottom of the window, it says "Activity: 5 -- ActiveCode (lcab2)".



# Check Your Understanding

1. Highlight the values at index 1 and 3 in the following array:

3	2	1	-3
---	---	---	----

2. What index is the first element in an array at?
- 0
  - 1

3. Highlight the values at index 0 and 2 in the following array

4	-2	8	7
---	----	---	---

4. Which index is the last element in an array called highScores at?
- highScores.length
  - highScores.length - 1



## For More Resources and to Check Answers

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