Topic 1.9: Array [Basics]

© Lauren Himbeault 2024

Learning Goals (Week 1):

- Identify data types based on value
- Map variables to the current values
- Perform basic operations on variables
- Create and use Java and user-defined methods
- Format Printed Output
- Obtain and process user input from the console
- Use booleans, conditionals, and compound conditionals correctly
- Select and implement different types of loops depending on scenario
- Use special String and Math operations
- **Successfully implement and manipulate java arrays**

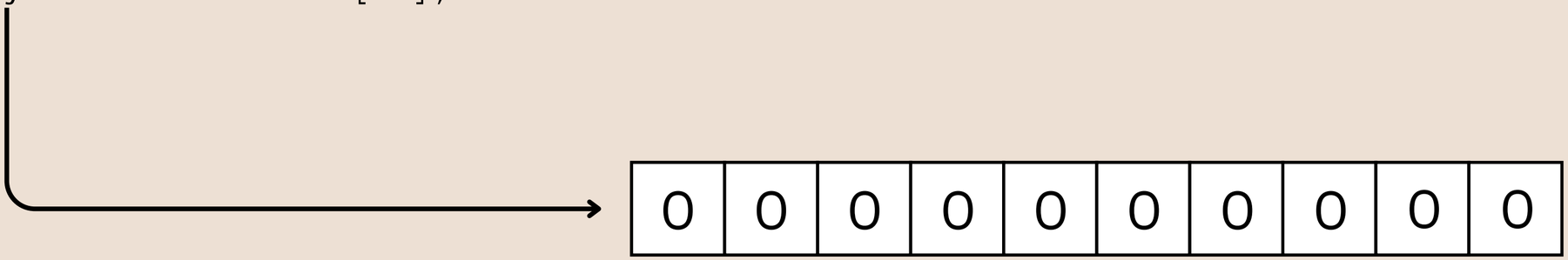
© Lauren Himbeault 2024

Arrays

- An array is an object that stores a group of values of the same type
- Can contain any number of elements (including 0)
- Once the size of the array is set, it cannot be changed (you have to make a new array and reset the whole thing to make a size change)

- you store the reference (memory address) of an array in a variable, **NOT THE ARRAY ITSELF** (all of this is true for any object, more on this to come)

```
int[] arrayOfInts = new int[10];
```



© Lauren Himbeault 2024

Arrays: More on Creation

- As previously shown, the new array is filled with default values for each type:

```
int [] -> 0
double[] -> 0.0
float[] -> 0.0f
boolean[] -> false
char[] -> ‘\u0000’
Any Object[], including String[] -> null
```

- Just like any variable, you can declare & instantiate in separate steps

```
int[] arrayOfInts; // equals Null
arrayOfInts = new int[10];
```

© Lauren Himbeault 2024

Arrays: More on Creation

- As previously shown, the new array is filled with default values for each type:

```
int [] -> 0
double[] -> 0.0
float[] -> 0.0f
boolean[] -> false
char[] -> ‘\u0000’
Any Object[], including String[] -> null
```

- Just like any variable, you can declare & instantiate in separate steps

```
int[] arrayOfInts; // equals Null
arrayOfInts = new int[10];
```

- You can hardcode initial values as well

```
int[] data1 = new int[]{1, 2, 3, 4, 5};

int[] data2;
data2 = new int[]{1, 2, 3, 4, 5};

int[] data3 = {1, 2, 3, 4, 5};  //OK to omit “new int[ ]”, only
                                //if you declare the variable on the same line
```

© Lauren Himbeault 2024

Arrays: Updating

- You can always change the value at a given position

```
int[] data1 = new int[]{1, 2, 3, 4, 5};

data1[0] = 7;                  // {7, 2, 3, 4, 5};
data1[1] = data[0] * data[1];

System.out.println(data[1]); // prints ???
```

© Lauren Himbeault 2024

Arrays: Length Property

- Getting the length of an array (number of cells) is easy: use myArray.length;

```
double[] myArray = new double[]{1.0, 2.5, 3.44};
System.out.println(myArray.length); //prints ???
```

```
int[] myArray2 = new int[]{1.0, 2, 4.44};
System.out.println(myArray2[myArray2.length-1]); //prints ???
```

- **NOTE:** The difference between *length()* of a String and *length* of an Array

```
double[] arr = new double[]{1.0, 2.0, 3};
System.out.println(arr);

String str = “Hello”;
System.out.println(str.length());
```

© Lauren Himbeault 2024

Arrays: Printing

- Unlike Python, you cannot just put your array variable in a print statement (it will print the reference…)

```
double[] arr = new double[]{1.0, 2.0, 3};
System.out.println(arr); // What does this print and why?
```

- You have to print the array yourself, by traversing the array using a loop:

```
for(int i=0; i < arr.length; i++) {

     System.out.print(arr[i]+" ");

}

System.out.println( ); //just adding a newline at the end
```

© Lauren Himbeault 2024

Arrays: Printing II

- You could also use the “Arrays” class (needs import statement)

```
import java.util.Arrays; //at top of file
...
...
...
   double[] arr = new double[]{1.0, 2.0, 3};
   System.out.println(Arrays.toString(data)); //returns a String representation of the array
```

© Lauren Himbeault 2024

Pause & Practice

Create an array of integers with a predefined set of values (e.g., int[] myArray = {1, 2, 3, 4, 5};).

Write a method to reverse the elements of this array.

The method should not return a new array; it should modify the original array.

After reversing, print out the elements of the modified array.

Example:

Input Array: *[1, 2, 3, 4, 5]*

Output Array after Reversal: *[5, 4, 3, 2, 1]*

© Lauren Himbeault 2024