Topic 1.10: Arrays [Advanced]

Learning Goals (Week 1):

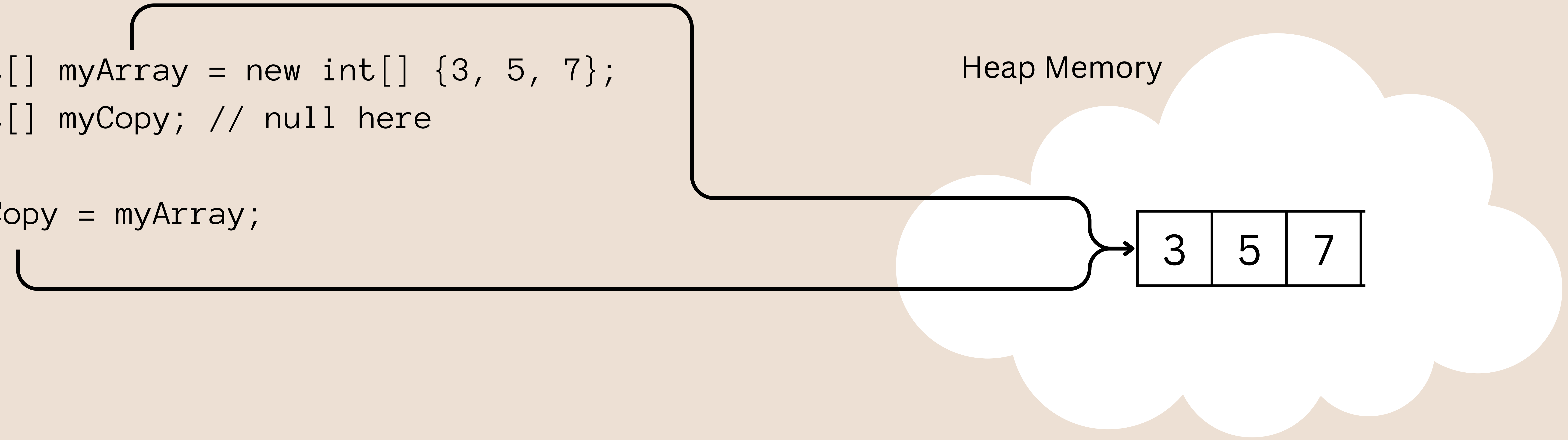
- Identify data types based on value
- Map variables to the current values
- Perform basic operations on variables
- Create and use Java and user-defined methods
- Format Printed Output
- Obtain and process user input from the console
- Use booleans, conditionals, and compound conditionals correctly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

Copying Arrays: Shallow Copy

- Here’s how **not** to copy an arrray:

```
int[] myArray = new int[] {3, 5, 7};
int[] myCopy; // null here

myCopy = myArray;
```

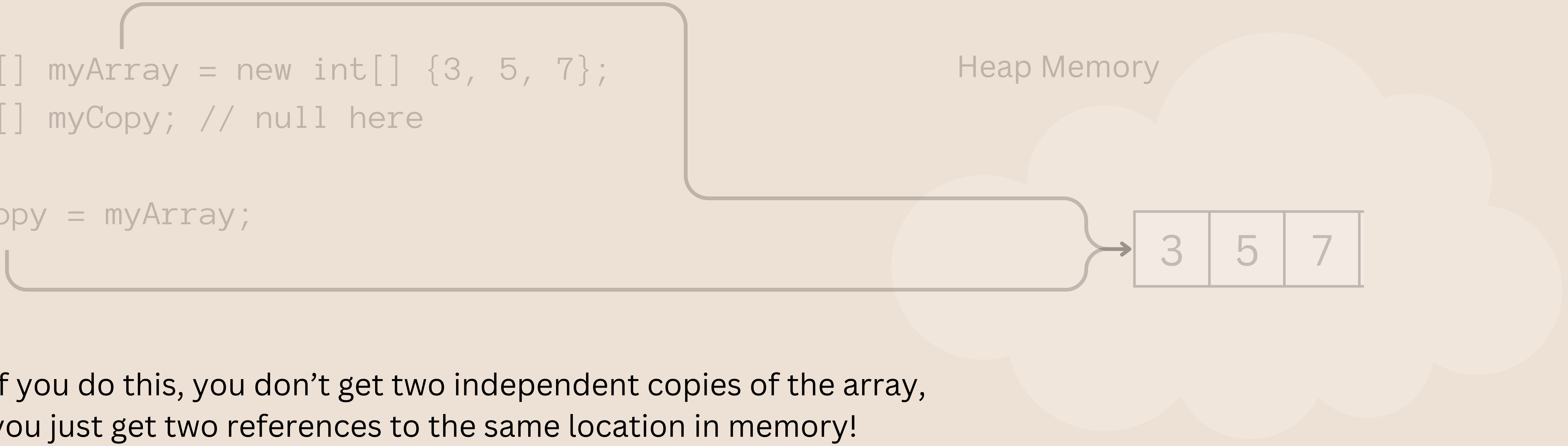


Copying Arrays: Shallow Copy

- Here's how **not** to copy an arrray:

```
int[] myArray = new int[] {3, 5, 7};
int[] myCopy; // null here

myCopy = myArray;
```



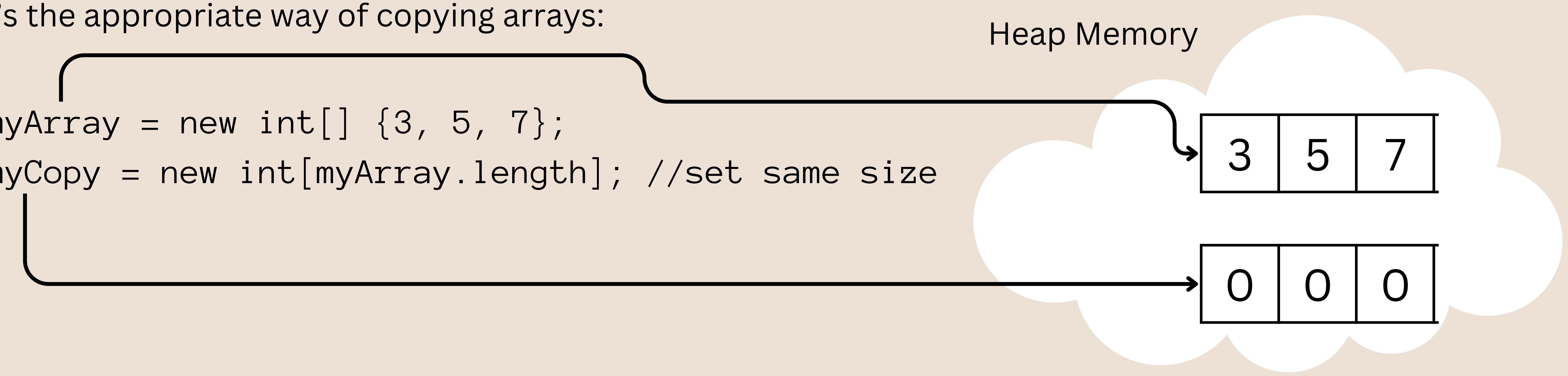
- If you do this, you don't get two independent copies of the array, you just get two references to the same location in memory!

- Modifying myArray's elements will also affect myCopy, because they both point to the exact same array in memory!

Copying Arrays: DeepCopy

- Here's the appropriate way of copying arrays:

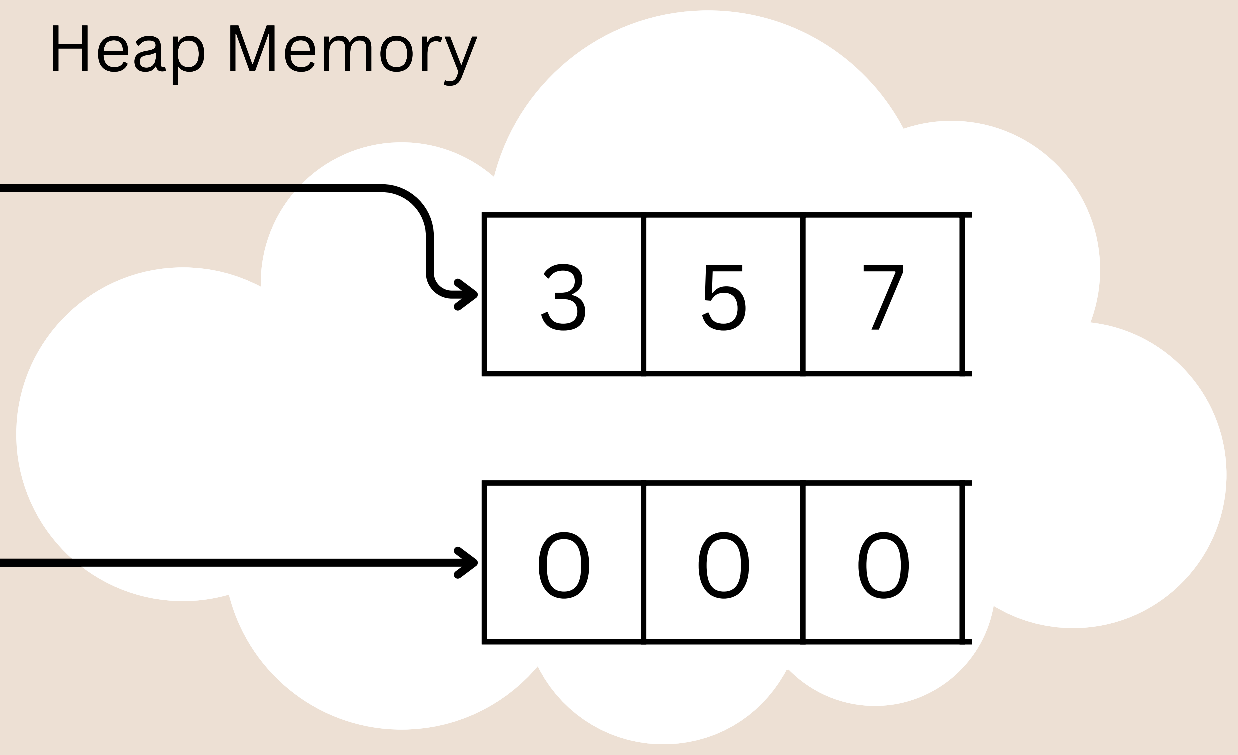
```
int[] myArray = new int[] {3, 5, 7};
int[] myCopy = new int[myArray.length]; //set same size
```



Copying Arrays: DeepCopy

- Here's the appropriate way of copying arrays:

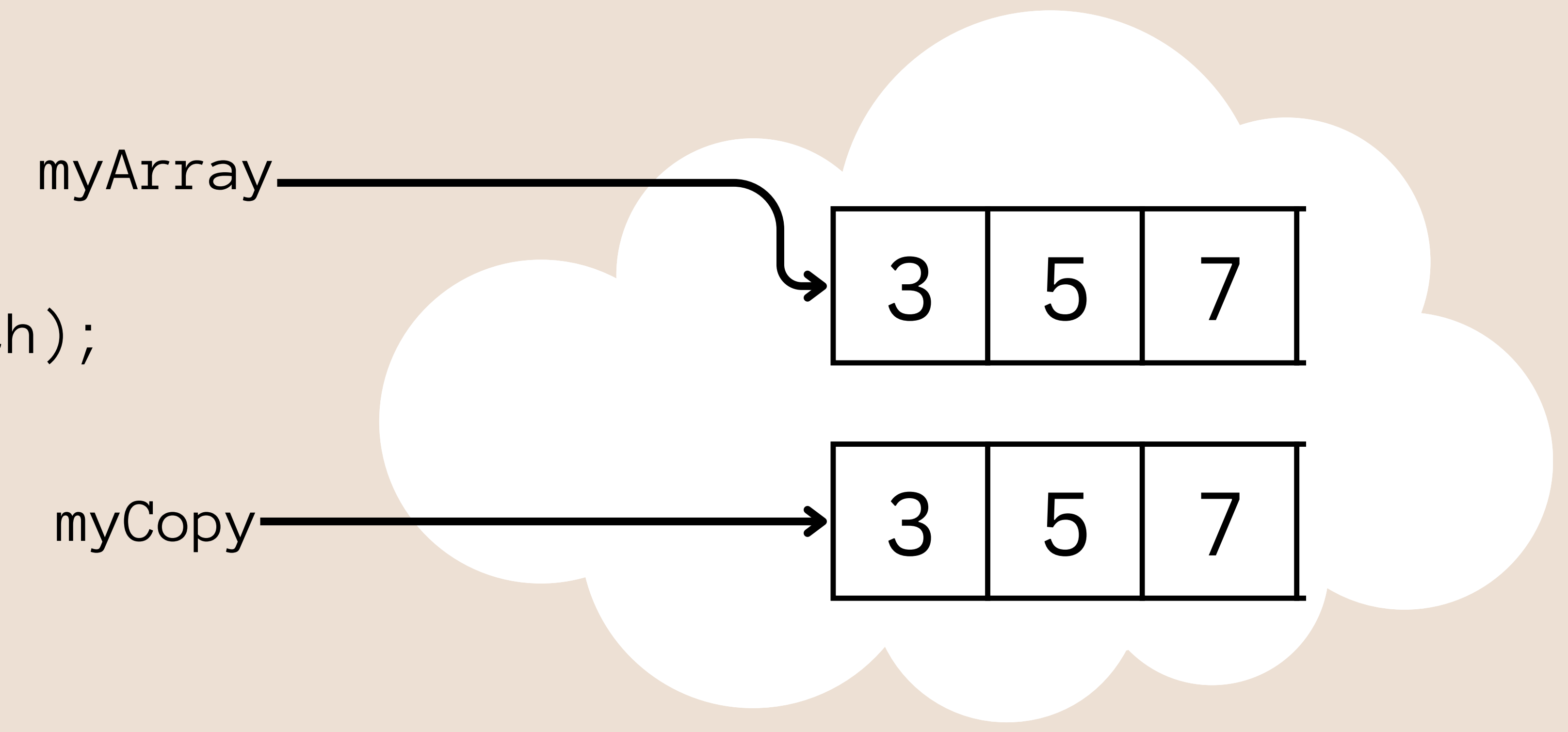
```
int[] myArray = new int[] {3, 5, 7};
int[] myCopy = new int[myArray.length]; //set same size
```



```
for (int i = 0; i < myArray.length; i++) {
   myCopy[i] = myArray[i];  //copies each element
}
```

- Alternative to using a for loop:

```
System.arraycopy(myArray, 0, myCopy, 0, myArray.length);
```



For Each Element: A Shortcut

- There’s a syntax shortcut for iterating over all elements in an array

```
for (int i = 0; i < data.length; i++) {
   System.out.println(data[i]);
}
```

- You can do instead

```
for (int element : data) { // for each element in data -> element = data[i]
   System.out.println(element);
}
```

Pause & Practice

1. **Create an Original Array:** Initialize an array of integers with a set of values (e.g., `{1, 2, 3, 4, 5}`).

2. **Copy the Array:**
   - Implement two methods to copy this array into a new array.
     - **Method 1:** Use a loop to manually copy each element.
     - **Method 2:** Use the `System.arraycopy()` method.

3. **Modify the Original Array:** After copying, modify one element in the original array (e.g., set the first element to `10`).

4. **Display Both Arrays:** Print both the original and the copied arrays to show that they are separate and that changing the original does not affect the copy.

5. **Discuss the Results:** Write a brief explanation of why the changes to the original array did not affect the copied array, emphasizing the concept of array references in Java.