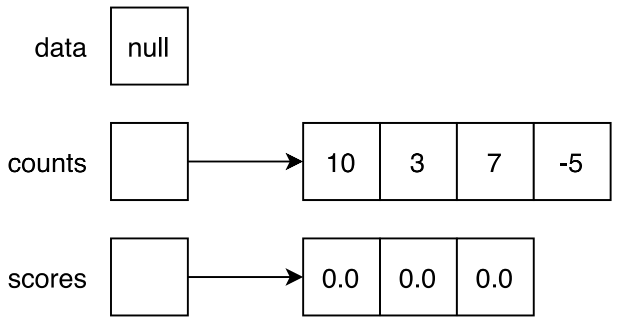
Model 1 Arrays

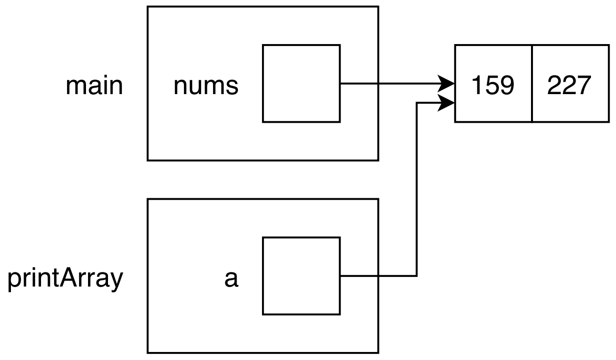
An array variable stores a ***reference*** to an array object. We draw references as arrows, because they "point" to other memory locations.

```
int[] data = null;

int[] counts = {10, 3, 7, -5};

double[] scores = new double[3];
```



When passing an array to a method, only the reference is copied:

```
public static void printArray(int[] a) {
    System.out.print("{" + a[0]);
    for (int i = 1; i < a.length; i++) {
        System.out.print(", " + a[i]);
    }
    System.out.println("}");
}

public static void main(String[] args) {
    int[] nums = {159, 227};
    printArray(nums);
}
```



Questions (15 min) **Start time:**

1. What is the length of each array?

a) `counts`?

b) `scores`?

c) `nums`?

d) `a`?

2. Looking at both diagrams above:

a) How many array variables were declared?

b) How many array objects were created?

3. Based on the top diagram, what is different about the variable named `data`?

4. Based on `counts` and `scores`, describe two ways that array objects can be created. How are these two ways different from each other?

5. If the `printArray` method were to modify the array contents, would that change be visible in the `main` method? Explain your reasoning.

6. Draw (or describe) a diagram of the following source code:

```java
int[] data = {1, 2, 3};
int[] copy = data;
```

7. (Optional) Paste the contents of *Arrays.java* into Java Visualizer. What differences do you notice between the diagram in Java Visualizer and those in Model 1?