

# Introduction to Programming

*with Java, for Beginners*

Arrays of Primitives  
Arrays of Objects

What if our Frog (from lab) could say 10 different things?

```
public class Frog{
    private boolean formerPrince;
    private String phrase1;
    private String phrase2;
    private String phrase3;
    private String phrase4;
    private String phrase5;
    private String phrase6;
    private String phrase7;
    private String phrase8;
    private String phrase9;
    private String phrase10;
    ...
}
```

1

What a Person could adopt lots of Pets ?

```
public class Person{
    private String name;
    private Pet pet1;
    private Pet pet2;
    private Pet pet3;
    private Pet pet4;
    private Pet pet5;
    private Pet pet5;
    private Pet pet6;
    private Pet pet6;
    private Pet pet8;
    private Pet pet9;
    private Pet pet10;
    private Pet pet11;
    private Pet pet12;
    private Pet pet13;
    private Pet pet14;
    private Pet pet15;
    ...
}
```

2

What if we want to store lots of things...

- But we don't want to declare a separate variable for each one?
- That's what *arrays* are good for

3

## What is an Array ?

- It's an easy way to declare lots of variables that all have the *same type*

```
type [] variableName = new type [#];
```

E.g. declare an array of integers  
`int[] data = new int[5]; //total ints = 5`

- When an array of particular primitive type is created, Java initializes the elements to the types default value. E.g. Array of ints – default value is zero

0	0	0	0	0
---	---	---	---	---

4

## Array Elements and Indices

- To initialize integer array

- `data[0] = 6;`
- `data[1] = 10;`
- `data[2] = 12;`

6	10	12	0	0
---	----	----	---	---

- The number within square brackets is called an *index*
- The valid *indices* are 0 thru (array length - 1)
  - 0 : first element of the array
  - n-1: last element of the array

5

## An Array is an Object

<code>int[] data;</code>	<i>data</i> is a reference variable whose <i>type</i> is <code>int[]</code> , meaning "array of ints". At this point its value is null.
<code>data = new int[5];</code>	The <i>new</i> operator causes a chunk of memory big enough for 5 ints to be allocated on the heap. Here, <i>data</i> is assigned a reference to the heap address.
<code>data[0] = 6;</code> <code>data[1] = 10;</code> <code>data[2] = 12;</code>	Initially, all five ints are 0. Here, three of them are assigned other values.
<code>int[] info = {6, 10, 12, 0, 0};</code>	
<code>int[] info = new int[]{6, 10, 12, 0, -1};</code>	

6

## Using Array Elements in Expressions

- An *element* of an array of ints can be used virtually anywhere an expression of type int is valid.
- Likewise for arrays of other types

```
int[] data = new int[] {6, 10, 12, 0, 0};  
int x = data[0];  
data[3] = data[2];  
data[4] = data[3] + data[2] * 2;  
System.out.println("data[0] is " + data[0]);  
data[4] = Math.pow(2, data[4]);
```

7

## Accessing an Array's Length

- `ArrayName.length` gives size of the array

```
int[] data;
data = new int[5];    // data.length is 5
data[0] = 6;
data[1] = 10;
data[2] = 12;

//How to Sum the contents of an array
int result = 0;
for (int i = 0; i < data.length; i++){
    result = result + data[i];
}
```

8

## Complete the sum(..) method

```
public class ArrayTool{

    /**
     * Takes an array of ints as an argument.
     * returns the sum of all the integers in the array.
     */
    public static int sum (int [] data ) {
        for (int i =0; i < data.length; i++){
            result = result + data[i];
        }
    }

    Welcome to DrJava
    > int[] data = new int[] {6, 10, 12, 0, 0};
    > ArrayTool.sum(data)
    28
}
```

9

## Array Out of Bounds Exceptions

```
public class ArrayTool{

    public static int sum(int[] data){
        int sum = 0;
        for (int i = 0; i <= data.length; i++){
            sum = sum + data[i];
        }
        return sum;
    }
}
```

```
> int[] data = new int[] {6, 10, 12, 0, 0};
> ArrayTool.sum(data)
ArrayIndexOutOfBoundsException
```

10

## Declaring & Initializing Arrays of Primitive Type

```
int[] info1 = { 2000, 100, 40, 60};
int[] info2 = new int[] {2000, 100, 40, 60};
```

```
char[] choices1 = { 'p', 's', 'q'};
char[] choices2 = new char[] { 'p', 's', 'q'};
```

```
double[] temps1 = {75.6, 99.4, 86.7};
double[] temps2 = new double[] {75.6, 99.4, 86.7};
```

*Note: The advantage of using the "new type[]" syntax is that it can be used in an assignment statement that is not a variable declaration statement.*

11

## Complete this method

```
public class ArrayTool{  
  
    /* Returns true if all integers in the  
    data array are positive, false otherwise.  
    */  
    public static boolean allPositive(int[] data){  
  
    }  
}
```

12

## Finding Max

- Complete max method, that finds the maximum value in the array
  - Assume that input passed is valid, no need for error checking

```
//Dr Java Interactions Pane  
> int[] data = new int[] {6, 10, 12, 0, 0};  
> ArrayTool.max(data)  
12
```

13

## Array of Primitives

```
int[] data;
```

data 

```
data = new int[3];
```

data  → 


```
data[0] = 5;  
data[1] = 10;
```

data  → 

14

## Array of Objects

```
Counter[] counters;
```

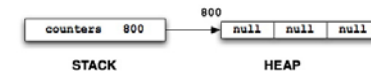


STACK

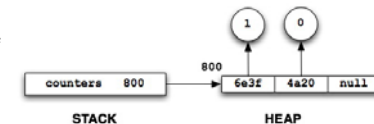
HEAP

```
counters = new Counter[3];
```

- it has 3 references to Counters in it
- all of these references are initially null



```
> counters[0]= new Counter();  
> counters[0].addOne();  
> counters[1]= new Counter();  
> counters[0].getCount()  
1  
> counters[2].getCount()  
NullPointerException
```



15

## E.g. Person Object

```
public class Person{
    private String name;
    private int age;

    Person(String name, int age){
        this.name = name;
        this.age = age;
    }

    public int getAge() { return age; }
    public String getName() { return name; }
}
```

16

## Person Array

```
Person[] people = {new Person("jo"),new Person("flo")};
Person[] people = new Person[] {new Person("jo"),
                                new Person("flo")};
```

Person[] people;

people null

people = new Person[3];

people 800 → null null null

```
people[0]=new Person("jo");
people[1]=new Person("flo");
```

people 800 → 6e3f 4a29 null  
jo flo

## E.g. Person Database (contd..)

```
public class PersonDB{
    private Person[] people;

    public PersonDB(){
        people = new Person [] {new Person("jo",25),
                                new Person("flo",18),
                                new Person("mo", 19)};
    }

    /** Calculates and returns the average age. */
    public double getAverageAge(){
        //Complete
    }
}
```

18

## In PersonDB.java

- Complete printDB()

```
>PersonDB people = new PersonDB();
>people.printDB()
Name Age
jo 25
flo 18
mo 19
```
- Complete isInDatabase(...) method

```
> PersonDB people = new PersonDB();
> people.isInDatabase("flo")
true
> people.isInDatabase("rex")
false
```

19