

## Introduction to Java Arrays Instructions:

### Explore and Discuss

Download and open the `Array_Example.jar` file. This program instantiates two arrays of integers in different ways. The arrays are then passed to an object for processing. Make an instance of an `Array_Driver` object and run the method `driver()` several times.

Open the `Array_Driver` class read all of the comments carefully. Pay particular attention to how the arrays are instantiated and how they are passed as parameters to the object. Now open the `Array_Processor` class. Study the implementation and comments of each method with your partner.

### Discuss and Answer

Answer the following questions with your partner and be prepared to discuss your thinking with the classroom:

1) Consider the following implementation:

```
int[] v = new int[8];
```

Which of the following best describes the state of array `v[]`?

- a. The elements `v[1] ... [8]` are initialized to zero.
- b. The elements `v[0] ... [7]` are initialized to "".
- c. The elements `v[1] ... [8]` are null.
- d. The elements `v[0] ... [7]` are null.
- e. The elements `v[0] ... [7]` are initialized to zero.

2) Consider the following implementation:

```
int[] k = new int[];
```

```
int f = k[4];
```

What is the best description of the result?

- a. This would cause a null pointer exception.
- b. This would cause a compile time error.
- c. `f = 0;`
- d. This would cause an index out of bounds exception.
- e. `f = 5;`

3) The contents of an array named `daps[]` are shown below:

```
{83, 61, 10, 94, 84}
```

`daps[0];` //evaluates to?

4) The contents of an array named `thowless[]` are shown below:

```
{91, 59, 44, 53, 77, 65, 37, 86, 42}
```

`thowless[1] + thowless[4];` //evaluates to?

5) The contents of an array named `hyphenates[]` are shown below:

```
{96, 76, 18, 94, 40, 22, 48, 35, 27}
```

```
hyphenates[5 + 1]; //evaluates to?
```

- a. 136
- b. 6
- c. 98
- d. This would cause an index out of bounds run-time error.
- e. 48

## Programming Challenge

Download and open the `Array_Stats_Assignment.jar`. This program generates an array of random numbers based on input from the user and then displays the array along with some calculated statistics. Make an instance of an `ArrayDriver` object and run the method `driver()`. Input a variety of parameters and observe the output. Open the `ArrayDriver` class and look for the reserved word `this`. Objects are capable of calling their own methods by referring to themselves using the implicit `this` reference. As in `this.someMethod()`. You likely have seen examples of `this` when working code-trace problems from the problem generator.

When you are ready, delete the `ArrayCalculator` class. Make a new class with the same name. Be sure to fill in the comment block at the top. Erase all of the code between the outermost `{ }`. You will need to make at least 5 methods. Documentation for the `ArrayCalculator` class is in the `README.TXT` file.

Calculating the standard deviation is a several step process. An example is provided explaining the steps involved in `CalculatingStandardDeviation.png`. Do not duplicate the implementation of the `meanFinder()` method in your code. Use the reserved word `this` to your advantage.

Use a calculator to test and verify your program output thoroughly. Create a jar file named `Last_Init_Array_Statistics.jar` with your work and submit it to the dropbox.