

**Lesson Objective:** Students examine the implementation of arrays in a `_Example` project and use the information to answer five multiple-choice questions about array implementation. In the `_Assignment` project the students write a `class` that uses an array of primitives to make statistical calculations.

**Length of Activity:** Two days (approximately 100 minutes)

**AP Topics:**

- IV. Standard Data Structures
  - E. Arrays
- V. Standard Operations and Algorithms
  - B. Searching
    - 1. Sequential

**Materials:** *\*.jar file links go here*

**Suggested Lesson Activity:** **Explore and Discuss**

There is a wealth of information about arrays in the comments of the `Array_Driver` class. Circulate to keep students focused on reading all of the comments carefully. Remind the students that they have used the concept of indexes when they utilized `String` methods. You may have an example ready like:

What would the following code display on the terminal window?

```
String j = "java is fun";  
System.out.print(j.substring(0,4)); [java]
```

**Answers for “Discuss and Answer”:**

1. e. The elements `v[0] ... [7]` are initialized to zero.

2. b. This would cause a compile time error because the length of the array `k[]` was not established when it was instantiated

```
3.      83      |  
          v  
indices:  0  1  2  3  4  
values; 83,61,10,94,84
```

```
4.  136 =   59   +   77  
indices:  0  1  2  3  4  5  6  7  8  
values; 91,59,44,53,77,65,37,86,42
```

```
5.    48 = hyphenates[6]  
indices:  0  1  2  3  4  5  6  7  8  
values; 96,76,18,94,40,22,48,35,27 ]
```

**Programming  
Challenge:**

Students will see examples of the implicit `this` object reference in the `ArrayDriver` class and are expected to use `this` rather than duplicate the implementation of the `meanFinder()` method in their code. The implicit reference `this` is fairly intuitive and the students have seen many examples of its use since Chapter 2 if they have been working code-trace problems from the problem generator.

Calculating the standard deviation is a several step programming process. An example is provided explaining the steps involved in `CalculatingStandardDeviation.png`. Students should compile and test each step separately. Because the calculation involves division, beware of integer division errors! Emphasize that the students use a calculator to test and verify their program output thoroughly.

Preliminary Edition