Practice Coding Assignment 11 : Arrays & ArrayLists

This assignment is entirely optional and designed to give you practice writing code and applying lessons and topics for the current module.

This homework deals with the following topics:

- Arrays
- ArrayLists
- Unit Testing

The Assignment

In this assignment, you will implement a class called *ArrayAndArrayList*. This class includes some interesting methods for working with Arrays and ArrayLists.

For example, the *ArrayAndArrayList* class has a "findMax" method which finds and returns the max number in a given array. For a defined array: int[] array = {1, 3, 5, 7, 9}, calling findMax(array) will return 9.

There are 4 methods that need to be implemented in the *ArrayAndArrayList* class:

- howMany(int[] array, int element) - Counts the number of occurrences of the given element in the given array.
- findMax(int[] array) - Finds the max number in the given array.
- maxArray(int[] array) - Keeps track of every occurrence of the max number in the given array.
- swapZero(int[] array) - Puts all of the zeros in the given array, at the end of the given array.

Each method has been defined for you, but without the code. See the javadoc for each method for instructions on what the method is supposed to do and how to write the code. It should be clear enough. In some cases, we have provided hints and example method calls to help you get started.

For example, we have defined a "howMany" method for you (see below) which counts and returns the number of occurrences of a given element in a given array. For now, the method returns a 0 value as a placeholder. Read the javadoc, which explains what the method is supposed to do. Then write your code where it says "// TODO" to implement the method. You'll do this for each method in the program.

```
/**
```

```
 * Counts the number of occurrences of the given element in the given
array.
 * @param array to search
 * @param element to search for
 * @return number of times element is in array
 */
public int howMany(int[] array, int element) {
    // TODO Implement method
    return 0;
}
```

In addition, you will write unit tests to test your method implementations. Each unit test method has been defined for you, including some test cases. First make sure you pass all of the provided tests, then write **additional and distinct test cases** for each unit test method.

For example, we have defined a "testHowMany" method for you (see below) which tests the "howMany" method. Pass the tests provided then write additional tests where it says "// TODO". You'll do this for each unit test method in the program.

```
/**
 * Test howMany method in ArrayAndArrayList.
 */
@Test
void testHowMany() {
    // element in the array
    int[] array = {1, 3, 5, 7, 9, 1, 2, 3, 4, 5};
    assertEquals(2, this.myArrayAndArrayList.howMany(array, 1));

    // TODO write at least 3 additional test cases
}
```

Tips for this Assignment

In this assignment, some tips are given as follows:

- Unit testing a method that doesn't return anything in Java:
    - Although a method doesn't return a value, it has some side effects and can be tested. There may be a way to verify that the side effects actually occurred as expected.

      For example: Consider a method that modifies a given array. The elements in the array will be changed after the method is called. Thus, an array can be created, and then the method can be called. After that, the same array can be tested for correctness.

Here's an example testing the "swapZero" method, which puts all of the zeros in a given array, at the end of the array. The method updates the array itself, so we need to call the method, then test the array directly.

```
//create array
int[] array = {0, 1, 0, 2};

//call swapZero method with array
this.myArrayAndArrayList.swapZero(array);

//test updated array by comparing to another test array
int[] testArray = {1, 2, 0, 0};
assertArrayEquals(testArray, array);
```

- Creating an empty array in Java:
  - For example:

```
int[] array = new int[0];
```

Submission

You have been provided with *ArrayAndArrayList.java* and *ArrayAndArrayListTest.java*. To complete the assignment, implement the methods in *ArrayAndArrayList.java*, making sure you pass all the tests in *ArrayAndArrayListTest.java*. Then write **at least 3 additional and distinct test cases** for each unit test method in *ArrayAndArrayListTest.java*. Do not modify the name of the methods in *ArrayAndArrayList.java* or the automated testing will not recognize it.

You will submit two files for this assignment: *ArrayAndArrayList.java* and *ArrayAndArrayListTest.java*. Make sure your program and the unit testing files run without errors! Submit the completed program using the steps outlined in the assignment in Coursera.

Evaluation

This assignment is evaluated, so that you can assess how well you understand the material. However, the evaluation of the assignment will not affect your course grade.

Points:

1. Does your code function correctly? (12 pts)
   - howMany(int[] array, int element) - 2 pts
   - findMax(int[] array) - 2 pts
   - maxArray(int[] array) - 4 pts
   - swapZero(int[] array) - 4 pts

2. Did you write at least 3 distinct and valid test cases for each test method? Do all of your tests pass? (12 pts)
    - testHowMany() - 3 pts
    - testFindMax() - 3 pts
    - testMaxArray() - 3 pts
    - testSwapZero() - 3 pts