Lab 7 (22 points total)

The purpose of this lab is to provide hands on practice for Chapter 7 concepts. There are several learning objectives to this assignment

- Writing Class and Method Definitions with a focus on Arrays.
- Using debugging techniques
- Incorporating documentation and style into your code

Lab 7 Part1 (11 points)

Use the following UML to create a class called ArrayMethods. ArrayMethods will need to use the java.util.Arrays class (requires import). **NOTE-Arrays copyOf() is a static method, meaning you do not need to create an object. For example, you can just do Arrays.copyOf()**

ArrayMethods
- myArray: int [ ] (initialized to {7, 8, 8, 3, 4, 9, 8, 7})
Lab 7 PART 1 - Methods + count (): int (1pt) + sum (): int (1pt) + average (): double (1pt) +findMax (): int (3pts) +findIndexOfMax (): int (3pts) +print(int [ ] a): void **//provided below**

Lab 7 PART 2 - Methods
+findLast (int key): int (3pts)
+findAll(int key): int [ ] (4pts)
+getArray(): int[ ] (1pt)
+copyArray (): int [ ] *(Note: Use Arrays.copyOf())* (1pt)
+reverseArray(int[ ] inArray): int[ ] (2pts)

1) count () returns the number of values that are in the array
   (Note: Use an enhanced for (for each) loop with a statement in this method to compute the number of values) vs .length //HINT – Just use a simple accumulator
2) sum () returns the sum of the values that are in the array
3) average () returns the average of the array as a double. **HINT-What does (double)sum()/count() do?**
4) findMax () returns the value of the largest integer in the array. **HINT: set max=myArray[0] and then run through myArray[ ], if a value is bigger, update max. You will do similar for findIndexOfMax(), also see pg 428.**
5) findIndexOfMax () returns the index of the value of the largest integer in the array

Other items of interest in ArrayMethods class

a) Use Arrays.copyOf() to create copyArray().

Submit javadoc screenshots for ArrayMethods.java (1pt) – NOTE: You might need to take a few screenshots. Make sure you look at the screenshots to ensure that each constructor and method has a description. Also, make sure all parameters and return methods have appropriate tags.

Submit GitHub screenshot (1pt)

```
 /**
  prints an int array, nicely formatted
  @param inArray int array to print
 */
public void print(int[ ] inArray)
{
   System.out.print("{");
   int i;
   // print elements before the last, separated by commas
   for (i = 0; i < inArray.length - 1; i++)
      System.out.print(inArray[i] + ", ");
   // print last element.  Careful here to handle length 0
   if (inArray.length > 0)
      System.out.print(inArray[i]);
   System.out.println("}");
}
```

Lab 7Pt2 (13pts)

6) getArray() is a simple getter for int[ ] myArray instance var.
7) copyArray() creates a copy of the myArray[ ] using the Arrays.copyOf(). HINT: copyOf requires two arguments - http://docs.oracle.com/javase/7/docs/api/java/util/Arrays.html

   NOTE: It was necessary to import java.util.Arrays into ArraysMethodsDemo to do this.

8) findLast (int key) returns the index of the last (right most) value based on the parameter that is passed in or -1 if the value 'key' is not found. **You will want to start at the right most index, for example, myArray.length-1 and work to the left (index[0])**
9) findAll(int key) creates and returns a new array containing the index(es) of every occurrence of a target value. Return an empty array of length 0 that contains nothing if the target value does not occur. Use the following array below as a print method to print the resultant findAll [ ]

   Hints: use 2 loops for this. The first counts how many times the target occurs. Next create a new array, to hold this many indexes. The second loop puts the indexes into the new array. Make sure after you put the value in the new array that you move to the next element.
10) reverseArray(int[ ] inArray) uses the inputted array to reverse the order and returns either the parameter array or another array produced in the method. **DO NOT USE Collections or ArrayList methods. You must create your own** 😊

   There are a few ways to do this. Here are a couple of hints.
   a. Start at index[0] and index[lenth-1] and swap values, you will need a temp var to do this. After each swap move the left (start) index one to the right and the right (end) index one to the left, until the left (start) index > the right (end) index and return the param int[ ] –or-
   b. Create a new array the same size as myArray[ ]. Copy the value index[length-1] to

the new array index [0] and move left to right from the original int [ ] and right to left in the new int [ ]. This is similar to when we reversed the words in Lab4. Then return the assign the new array to the param var so that the array passed in now points to the memory location of the new int [ ].

Submit GitHub screenshot (1pt)

Submit javadoc screenshots for ArrayMethods.java (1pt) – NOTE: You might need to take a few screenshots. Make sure you look at the screenshots to ensure that each constructor and method has a description. Also, make sure all parameters and return methods have appropriate tags.

Submitting your work

For all labs you will need to provide a copy of all .java files. In addition to your .java files, you will need to provide output files of your console. The name of the output file should match the class name and have the .txt extension such as TempProbOut.txt, ProChall3Output.txt. For GUIs such as JOptionPane, you will instead need to create screenshots. For Windows users, Snipping Tool is a great way to do this. Mac OS users, you can see how to take screenshots using the following url - https://support.apple.com/en-us/HT201361.