# Assignment 6: Array Statistics

Instructions

For this assignment, you will complete two separate classes which will be used to find statistics from arrays.

The first class, **NumStatsArray**, needs to store an array of doubles as a member variable. This should be declared as private and final, so none of your methods will set this variable to point at a different array. The single constructor for this class has a parameter of a double array, and should set the member variable to point to this parameter array.

You will need to complete the following methods in the NumStatsArray class (you may assume that the member variable array is not empty when writing these methods):

- **toString** - returns a String with the array contents in the form {1.0, 4.0, 5.0}. Note that there is a comma then a single space between each pair of elements, but there are no spaces/commas at the start/end, only curly braces.
- **average** - returns the arithmetic mean average of the numbers in the array as a double. This is found by dividing the sum of the numbers by the amount of numbers in the array *(note: some values returned may be slightly inaccurate due to roundoff errors, which you do not need to worry about for this assignment)*.
- **range** - returns a double found by subtracting the smallest number in the array from the largest number in the array
- **sortStatus** - returns an int which is equal to 1 if the array is sorted in increasing order, -1 if it is sorted in decreasing order or 0 if it is not sorted. The array is sorted in increasing order if every number is greater than or equal to the previous number. The array is sorted in decreasing order if every element is less than or equal to the previous number. If neither of these is true then the list is unsorted. If every element in the array is the same, then it is counted as being in increasing order.

The second class, **StringStatsArray**, needs to store an array of Strings as a member variable. This should be declared as private and final, so none of your methods will set this variable to point at a different array. The single constructor for this class has a parameter of a String array, and should set the member variable to point to this parameter array.

You will need to complete the following methods in the StringStatsArray class:

- **toString** - returns a String with the array contents in the form {"string 1", "string 2", "string 3"}. Note that there are double quotes around each string, and that there is a comma then a single space between each pair of elements, but there are no spaces/commas at the start/end, only curly braces.

- **averageLength** - returns the arithmetic mean average of the length of the Strings in the array as a double *(note: as before some values returned may have roundoff errors, which you do not need to worry about for this assignment)*.
- **search** - has a single String parameter and returns an int which is equal to the index of the first appearance of the parameter String, or -1 if it does not appear in the array.
- **sortStatus** - returns an int which is equal to 1 if the array is sorted in alphabetical order, -1 if it is sorted in reverse alphabetical order or 0 if it is not sorted in either of these ways. Note that a String may appear twice in a row in an array which is sorted in alphabetical or reverse-alphabetical order. If every String in the array is the same, then it is counted as being in <u>alphabetical order</u>.

You should use the runner_StatsArray class to test your classes. Do not add a main method to the NumStatsArray.java or StringStatsArray.java files, as it will cause your submission to be scored incorrectly.

Milestones

As you work on this assignment, you can use the milestones below to inform your development process:

Milestone 1: Add the double array member variable to the NumStatsArrray class and complete the constructor. Write the toString method for the class and test that it formats the output string in the exact format described.

Milestone 2: Write a loop in the average method which sums the values, then use this to return the average. Write a loop to iterate through the array and find the minimum/maximum value in the range method, then use these values to calculate and return the range.

Milestone 3: Write code in the sortStatus method which iterates through the array. Use separate flag variables to determine at the end of the array whether it is in increasing or decreasing order. Add return statements which return the correct values depending on if/how the list is sorted.

Milestone 4: Add the String array member variable to the StringStatsArrray class and complete the constructor. Write the toString method for the class and test that it formats the output string in the exact format described.

Milestone 5: Reuse and modify the code used for the average method in the NumStatsArray class to implement the averageLength method in StringStatsArray. Remember that the lengths of Strings are ints not doubles, but your method must return a double for the average length. Write a simple linear search algorithm in the search method to return the position of the parameter string, or -1 if it does not appear.

Milestone 6: Reuse and modify the code used for the sortStatus method in the NumStatsArray class to write the sortStatus method for the StringStatsArray class. You may need to revise how to test whether one string appears before or after the other alphabetically.

NumStatsArray.java

StringStatsArray.java

runner_StatsArray.java

```java
public class NumStatsArray{

  private final double[] arr;

  public NumStatsArray(double[] arr){
    this.arr = arr;
  }

  public String toString(){
    String str = "{";
    for (int i = 0; i < arr.length; i++)
      double item = arr[i];
      if (i != arr.length-1) {
        str += item + ", ";
      } else {
        str += item;
      }
    }
    str += "}";
    return str;
  }

  public double average(){
    double sum = 0;
    for (double item : arr) {
      sum = sum + item;
    }

    return sum / arr.length;
  }

  public double range(){
    double min = arr[0];
    for (int i = 1; i < arr.length; i++)
      if (min > arr[i]) {
        min = arr[i];
      }
    }
    double max = arr[0];
    for (int i = 1; i < arr.length; i++)
```

For this assignment, you will complete two separate classes which will be used to find statistics from arrays.

The first class, **NumStatsArray**, needs to store an array of doubles as a member variable. This should be declared as private and final, so none of your methods will set this variable to point at a different array. The single constructor for this class has a parameter of a double array, and should set the member variable to point to this parameter array.

You will need to complete the following methods in the NumStatsArray class (you may assume that the member variable array is not empty when writing these methods):

- **toString** - returns a String with the array contents in