Lab 11 – Linear Search, Method Overloading, and Command Line Arguments

Introduction

One way of searching for an item in a one-dimensional array is called linear (or sequential) search. The purpose of this lab is to use linear search to locate the first index of an item in an input array **x**. If the item is not in the input array **x**, then a message from your program should indicate this. This lab will NOT use the Scanner class to accept input from the user. Instead, the inputs for this lab will be command line arguments where the first command line argument is -s (String) or -d (double) to denote the type of item and type of values found in array **x**, the second command line argument will be the item to search for, the third command line argument will be -x to denote an input array **x**, and the remaining command line arguments are the values to be put into the array **x** in sequential order. Four examples are below. In each example, the program is called with command line arguments on the fist line, and the second line is the output from a correctly implemented program.

```
java LinearSearch -s zoey -x gracie roddie Zoey sasha zoey joe zoey darla
zoey found in x at index 4

java LinearSearch -s toph -x sokka kitara aang zuko
toph not found in x

java LinearSearch -d 7.5 -x 9.73 10 12 7.5000 12.6 7.5 99 -28.5 7e10
7.5 found in x at index 3

java LinearSearch -d -77 -x 18 23 91.7 23.5 87 92 101.3 68.2
-77.0 not found in x
```

Lab Objectives

By the end of the lab, you should have gained experience implementing linear search with 1D arrays, overloading methods, working with command line arguments, and using a wrapper class.

What to Submit

The **`LinearSearch.java`** file should be submitted to our course website for grading.

Instructions

1. Create a class called **`LinearSearch`**. Include the standard header comment stating your name, the date, program purpose, and containing the statement of academic honesty. When writing, you must also abide by the Java formatting and style conventions.
2. You cannot use the Scanner class for this assignment (the use of the Scanner class or any of its methods will result in a grade of 0 on this assignment). Instead use command line arguments as inputs.
3. In your class, implement the following static methods, and comment all methods.
   a. **`public static int getFirstIndex(double item, double[] x)`** - If the **item** is in the array **x**, then return the first index **i** of **x** where **x[i]** is equal to **item**. Otherwise, return -1 if the **item** is not in the array **x**.
   b. **`public static int getFirstIndex(String item, String[] x)`** - If the **item** is in the array **x**, then return the first index **i** of **x** where **x[i]** is equal to (case sensitive) **item**. Otherwise, return -1 if the **item** is not in the array **x**.

   c. **`public static void main(String[] args)`** – The main method that process the **args** and calls **getFirstIndex** to get the correct output based on the examples found in this lab. You may assume the command line arguments follow the aforementioned pattern and the examples. Also, you may want to use the Double wrapper class to convert a String value to a double value when -d is an input.

4. Your program's I/O must match the examples in this lab and the further examples provided in the file **Lab11Examples.txt** when run on our Unix machines (note: this text file should be viewed with word wrapping disabled so long example inputs appear on a single line).

Submission and Grading

After you have completed and thoroughly tested your program, upload and submit **`LinearSearch.java`** to our course webpage in order to receive credit for the lab (if you have any questions about how to submit this assignment, then ask your lab instructor for help at least 24 hours before this assignment is due). Always double check that your submission was successful on our course website.

The lab will be graded according to the following guidelines.

- A score between 0 and 100 will be assigned.
- If the source file(s) are not submitted before the specified, if they do not compile, or if the submitting student has an unexcused absence for a single lab period devoted to the assignment, then a grade of 0 will be assigned. (Note: students who show up to their first lab period and ***personally*** show their lab TA that they finished and submitted the week's lab assignment may be excused from their second lab period for that week.)
- If the required comment for all labs describing the program and the academic honesty statement is not included at the top of the file, then 10 points will be deducted. Note: this required comment can be found in Lab 02.
- If the source code uses the Scanner class in any way, then a 0 will be assigned.
- The program will be evaluated using a multiple test cases.