COE528 (Winter 2024) - Lab2

General Lab Rules

- All the necessary files of this lab should be in lab2 directory.
- All the java files in this lab should have the following package declaration:

```
package coe528.lab2;
```

Duration: one week.

Objective: Implement and specify procedures with Requires, Modifies and Effects clauses.

The specification for a procedural abstraction contains:

(a) **Requires:** This clause states any constraints under which the procedure will work. This is an optional clause.
(b) **Modifies:** This clause lists the names of any inputs that are modified by the procedure. This is an optional clause.
(c) **Effects:** This clause describes the behaviour of the procedure for all inputs that are not ruled out by the Requires clause.

Ex1: Specification for `substituteMax` procedure

In the lab2 directory, create a text file named Ex1.txt. Copy the code provided below for the `substituteMax` method to the file Ex1.txt. Next, in the file Ex1.txt, write the specification (i.e. provide the necessary clause(s) out of the three clauses Requires, Modifies, Effects as relevant) corresponding to the code for `substituteMax` method.

```
//Requires: <Write the Requires clause here>
//Modifies: <Write the Modifies clause here>
//Effects:  <Write the Effects clause here>

public static void substituteMax(int[] a, int[] b) {
    int maxOfA = a[0];
    int index = 0;
    for (int i = 0; i < a.length; i++) {
        if(a[i] > maxOfA) {
            maxOfA = a[i];
            index = i;
        }
     }
    int maxOfB = b[0];
    for (int i = 0; i < b.length; i++) {
        if(b[i] > maxOfB) {
            maxOfB = b[i];
        }
     }
    a[index] = maxOfB;
}
```

Ex2: Specification and Implementation for `isPalindrome` procedure

In the Netbeans program, click on Project > New Project and save it as "Ex2" on your lab2 directory.

Create a new class called `Palindrome`. In this class:

1. Specify (i.e. provide the necessary clause(s) out of the three clauses Requires, Modifies, Effects as relevant) and implement a procedure called `isPalindrome` that determines whether or not a string is a palindrome. (A palindrome reads the same backward and forward; an example is "deed".) This method should be public and static, and take a single string parameter as follows. The method should return true if the specified string is a palindrome and false otherwise.

```
//Requires: <Write the Requires clause here>
//Modifies: <Write the Modifies clause here>
//Effects:  <Write the Effects clause here>
public static boolean isPalindrome(String a) {
      //write the code for isPalindrome

}
```

2. **Copy** the `main` method for the class `Palindrome` as follows:

```
public static void main(String[] args) {
     if(args.length == 1) {
         if (args[0].equals("1"))
             System.out.println(isPalindrome(null));
         else if (args[0].equals("2"))
             System.out.println(isPalindrome(""));
         else if (args[0].equals("3"))
             System.out.println(isPalindrome("deed"));
         else if (args[0].equals("4"))
             System.out.println(isPalindrome("abcd"));
     }
 }
```

If the `main` method is run with the command line arguments, the console should show relevant output. Each command line argument and the corresponding output is shown below.

Command line argument	Output
1	False
2	False
3	True
4	False

Ex3: Specification and Implementation for `Anagram Checker`

Implement a method in the AnagramChecker class to determine whether two strings are anagrams of each other. Save the AnagramChecker class with the implemented areAnagrams method in your lab project directory as "Ex3.java."

Class: AnagramChecker

1. Implement the areAnagrams method to check whether two given strings (str1 and str2) are anagrams.
2. The method should return true if the strings are anagrams and false otherwise.
3. Include the Requires, Modifies, and Effects clauses within the comments of the method.
4. Ensure that the method is case-insensitive and considers spaces. For example, "Listen" and "Silent" should be considered anagrams.
5. Test the implemented method by running a main method with command line arguments representing pairs of strings.

Test Cases:

- Run the program with command line arguments:
    - "listen" "silent" (Expected Output: true)
    - "Hello" "World" (Expected Output: false)
    - "Dormitory" "Dirty room" (Expected Output: true)

Submission

Deadline: The submission deadline is 11:59 PM EST, the night before the corresponding lab session during the week of February 5, 2024.

You must include the duly filled and signed standard cover page with your submission. The cover page can be found on the departmental web site: Standard Assignment/Lab Cover Page

You must submit your lab electronically on D2L. Please zip up your NetBeans project containing all source files and submit it to the respective assignment folder on D2L.