CSC 1302 Principles of Computer Science II

Lab 14: ArrayList Practice

(Due on 11:59 pm, 7/23/2021)

Purpose:

The ArrayList class is the part of Java Collection Framework. This class implements the List interface provided by the Collection framework. Here are features or key points about the ArrayList in Java:

1. The ArrayList grow and sink dynamically when we add/remove elements from it.
2. ArrayList use the array as an internal data structure to store element.
3. This provides flexibility to use elements using the indexes.
4. ArrayList allows to store duplicate values including “null” values.
5. It is an ordered collection i.e. it maintains the insertion order.
6. Java ArrayList supports Generics.
7. It only allows objects in the ArrayList. For primitives values like int, long etc. use the wrapper classes.
8. Java ArrayList Iterator and ListIterator implementation is fail-fast.
9. ArrayList is not thread safe. We need to apply synchronization if multiple threads will change ArrayList at the same time.

Task:

Write a program called ArrayList_Practice. In this program, please do the following operations.

1. Complete a method called RedundantCharacterMatch(ArrayList<Character> YourFirstName): the parameter of this method is an ArrayList<Character> whose elements are the characters in your first name (they should be in the order appear in your first name, e.g., if your first name is bob, then the ArrayList<Char> includes ‘b’, ‘o’, ‘b’.). The method will check whether there exists duplicate characters in your name and return the index of those duplicate characters. For example, when using bob as first name, it will return b: 0, 2.
2. Create an ArrayList<Character> NameExample. All the characters of your first name will appear twice in this ArrayList. For example, if your first name is bob, then NameExample will include the following element {b,o,b,b,o,b}. Then, please use NameExample as parameter for the method RedundantCharacterMatch(). If your first name is bob, the results that print in the console will be
   b: 0, 2, 3, 5
   o: 1, 4

Criteria:

1. Upload all of the .java and the .class files to the CSc1302 dropbox on http://icollege.gsu.edu.
2. Your assignment will be graded based on the following criteria: (a) Are your programs runnable without errors? (b) Do your programs complete the tasks with specified outputs? (c) Do you follow the specified rules to define your methods and programs? (d) Do you

provide necessary comments include the programmer information, date, title of the program and brief description of the program.

3. Please comment the important lines in the .java file as shown in the template. The important lines including but not limited to i) variables, ii) for-loop, iii) while-loop, iv) if-else statement, iv) methods. Please use your own words to describe what is your purpose to write this line. A .java file without comment will be graded under a 40% penalty.
4. Make sure that both the .java and .class files are named and uploaded to icollege correctly. If any special package is used in the program, be sure to upload the package too. Should you use any other subdirectory (whatsoever) your program would not be graded, and you will receive a **0 (zero)**.
5. No copying allowed. If it is found that students copy from each other, all of these programs will get **0**.

Considering the difficulty of this lab, the code is listed below; when you cannot come up with a solution, please try to understand the code below and write comments.

```
package labs;

import java.util.ArrayList;
import java.util.Arrays;

public class Lab15 {
        public static void main(String[] args){
                ArrayList<Character> name = new ArrayList<Character>();
                name.add('b');
                name.add('o');
                name.add('b');
                ArrayList<Character> name1 = new ArrayList<Character>();
                name1.add('b');
                name1.add('o');
                name1.add('b');
                name1.add('b');
                name1.add('o');
                name1.add('b');
                String a = "b";
                RCM(name1);
        }
        public static void RCM(ArrayList<Character> name){
                ArrayList<String[]> Temp = new ArrayList<String[]>();
                for(int i = 0; i<name.size();i++){
                        int judge = CheckNumber(Temp, name.get(i));
                        if(judge== -1){
                                String[] newString = new String[2];
                                newString[0] = name.get(i) +"";
                                newString[1] = i +"";
                                Temp.add(newString);
                        }
                        else{
                                String[] tempString = Temp.get(judge);
                                tempString[1] = tempString[1] + i;
                                Temp.set(judge, tempString);
                        }
                }
                for(String[] x : Temp)
                        System.out.println(Arrays.toString(x));
        }

        public static int CheckNumber(ArrayList<String[]> Temp, char a){
                for(String[]x:Temp){
                        if(x[0].matches(a +"")){
                                return Temp.indexOf(x);
                        }
                }
                return -1;
        }

}
```