# Xi’an JIAOTONG-LIVERPOOL UNIVERSITY

西 交 利 物 浦 大 学

# Coursework Submission Cover Sheet

|  |  |  |
| --- | --- | --- |
| Name | Jin(Surname) | Minhao(Other Names) |
| Student Number | 1717576 | |
| Programme | Information and Computing Science | |
| Module Title | Data Structures | |
| Module Code | CSE104 | |
| Assignment Title | Assignment 1 | |
| Submission Deadline | Friday, 19 April 2019, 12:00 PM | |
| Module Leader | Prof. Steven Guan | |

By uploading or submitting this coursework submission cover sheet, I certify the following:

* I have read and understood the definitions of PLAGIARISM, COLLUSION, and the FABRICATION Of DATA, as outlined in the Undergraduate Student Handbook of Xi’an Jiaotong-Liverpool University and as posted on the University Website.
* This work is my own, original work produced specifically for this assignment. It does not misrepresent the work of another person or institution as my own. Additionally, it is a submission that has not been previously published, or submitted to another module.
* This work is not the product of unauthorized collaboration between myself and others.
* This work is free of embellished or fabricated data.

I understand that PLAGIARISM, COLLUSION and the FABRICATION OF DATA are serious disciplinary offences. By uploading or submitting this cover sheet, I acknowledge that I am subject to disciplinary action if I am found to have committed such acts.

Signature ……*.*…………………………………… Date ……*………………………………*……………

|  |  |  |  |
| --- | --- | --- | --- |
| For Academic Office use: | Date Received | Days Late | Penalty |
|  |  |  |

## Feedback on the strength of the work

Feedback on the weakness that needs to be improved

**1st Marker Date Mark 2nd Marker Date Mark**

**(if applicable)**

***Students:*** *Please start your assignment on the next page.*

The structure of NewList:

public class NewList<E> extends ArrayList<E> {

private Object[] data; //’data’ is used to store the input array

private int manyItems; //’manyItems’ is to record the data’ s length

public NewList(E[] data) { //constructor

this.data = data;

manyItems = data.length;

}

}

Task 1 pseudocode:

public < E > int remove1(E target) {

for index=0 to manyItems do

begin

if data[index] == target then data[j] = data[manyItems-1]

manyItems--

index--

end

data = Arrays.copyOf(data, manyItems);

return data.length

}

Task 2 pseudocode:

public < E > int remove2() {

Map<Object, Integer> helper = new HashMap

for each Object o in data do

begin

helper.put(o, helper.getOrDefault(o, 0) +1)

end

List<Object> target = new ArrayList

for each Map.Entry<Object, Integer> entry in helper.entrySet do

begin

if entry.getValue() == 3 then

target.add(entry.getKey())

end

for each Object o in target do

begin

for index = 0 to manyItems do

begin

if data[index] == o then

for j = index to manyItems – 1do

begin

data[j] = data[j + 1];

end

manyItems--;

index--;

end

end

data = Arrays.copyOf(data, manyItems)

println("The new List: " + Arrays.toString(data))

return data.length

}

Task 3 pseudocode:

public < E > List<List<String>> anagrams() {

String[] str = new String[data.length]

for i = 0 to data.length do

begin

str[i] = data[i].toString

end

if data.length == 0 then

return new ArrayList

Map<String, Set> helper = new HashMap<String, Set>

for each String s in str do

begin

char[] ca = s.toCharArray

Arrays.sort(ca)

String K = String.valueOf(ca)

if !helper.containsKey(K) then

helper.put(K, new HashSet())

helper.get(K).add(s)

end

List<List<String>> res = new ArrayList

for each Set set in helper.values do

begin

List<String> list = new ArrayList(set)

if list.size() > 1 then

res.add(list)

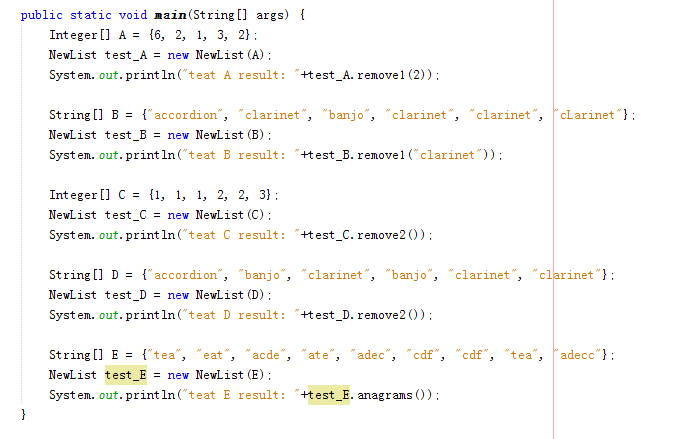
end

return res;

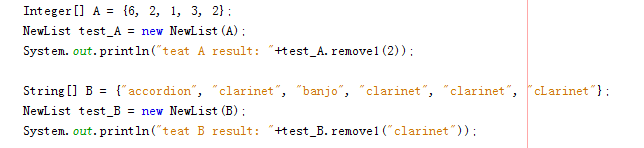
}

Snap shot:

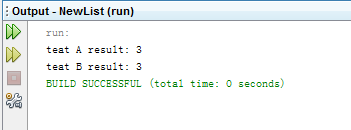
The main method for testing:



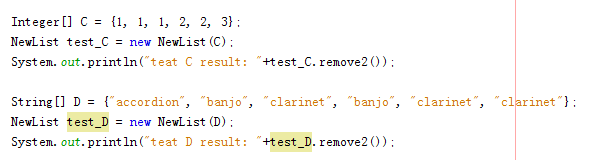
Task 1 testing:



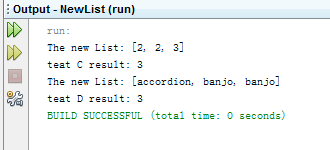
Task 1 testing results:



Task 2 testing:



Task 2 testing results:



Task 3 testing:



Task 3 testing results:

