Instructions for Practical Exercise

1. Use the Java_language_basics folder of previous part. This folder will contains code files for each PE that you will do for the Java Inheritance and Common Classes PE.
2. Name code files corresponding to PE numbers. For example, Pe1.java is the code file for Practical Exercise 1 (PE 1)
3. Push your project to git

Documentation Exercise

Document the difference between Set, List, and Map.

Document scenarios when to use which type of collections.

Note: Save the document as Collection.md in a doc folder of your project. Use markdown language https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet

Practical Exercise: Java Inheritance and Common Classes – Part 4

PE 1: Write a Java program that accepts an array and converts it to an ArrayList

PE 2: Write a program to find the number of counts in the following String. Store the output in Map<String,Integer> as key value pair.

Input : String str = "one one -one___two,,three,one @three*one?two";

Output : {"one":5 , "two":2, "three" :2}

PE 3: Write a program where an array of strings is input and output is a Map<String,boolean> where each different string is a key and its value is true if that string appears 2 or more times in the array

Input : String arr[] = {"a","b","c","d","a","c","c"}

Output - {"a" : true,"b" :false ,"c" :true,"d" : false}

PE 4: Create a Student class that represents the following information of a student: id, name, and age all the member variables should be private .

a. Implement `getter and setter` .

b. Create a `StudentSorter` class that implements `Comparator interface` .

c. Write a class `Maintest` create Student class object(minimum 5)

d. Add these student object into a List of type Student .

e. Sort the list based on their age in decreasing order.

For student having same age, sort based on their name.

f. For students having same name and age, sort them according to their ID.

PE 5: Write a method that accepts a Map object having two key-value pairs with the keys val1 and val2. Modify and return the given map as follows:

a. If the key `val1` has a value, set the key `val2` to have that value, and

b. Set the key `val1` to have the value `"  "` (empty string).

Example 1:

The map {"val1": "java", "val2": "c++"} should return {"val1": " ", "val2": "java"}

Example 2:

The map {"val1": "mars", "val2": "saturn"} should return {"val1": " ", "val2": "mars"}

PE 6: Write a program to implement Set interface which sorts the given randomly ordered names in ascending order. Convert the sorted set in to an array list

Input : Harry Olive Alice Bluto Eugene

Output :

Sorted Set : Alice Bluto Eugene Harry Olive

Array list from Set : Alice Bluto Eugene Harry Olive

PE 7: Complete the following program that compares two ArrayList objects. The program should take an element of the first ArrayList and check whether the element is present in the second ArrayList. The program should then store the comparison results in a separate ArrayList object as either "Yes" or "No" values.

Partial Code to be completed

```
import java.util.*;
  public class Exercise13 {
  public static void main(String[] args) {
   ArrayList<String> c1= new ArrayList<String>();
```

```
            c1.add("Java");
            c1.add("C");
            c1.add("C++");
            c1.add("Python");
            c1.add("JavaScript");

            ArrayList<String> c2= new ArrayList<String>();
            c2.add("Java");
            c2.add("Angular");
            c2.add("C++");
            c2.add("JavaScript");

            //Complete code to store the comparison output in
ArrayList<String>

     }
}
```

Expected output:

[Yes, No, Yes, No, Yes]