Lab 01: Java Collections

CO324: Network
and
Web Application Design

January ~~10~~ 13, 2017

1 Goals

By the end of this lab, you should be able to:

Write simple Java programs using

Maps and

Lists

2 Introduction

We try to identify, given a specific scenario, what construct ideally suits a problem at hand. Before proceeding further, we first analyze the properties of the above few important data arrangements.

3. Hash Map

A hash map identifies a value, by means of a key - like a course code fetching its title.

Basic operations are:

```
put(), get(), remove(), getKey(), getValue()
```

Declarations:

```
HashMap<Integer, String> hmap = new HashMap<Integer, String>();
```

Iteration:

```
Set set = hmap.entrySet();
Iterator iterator = set.iterator();

while(iterator.hasNext())
```

More recommended:

```
for (Map.Entry<String, String> entry : hmap.entrySet()) {
```

Ex: Write a Hash Map that would return some A-grade roads, when you specify its number:

For example A2 -> Colombo - Wellawaya
A5 -> Peradeniya – Chenkalady
A3 -> Peliyagoda - Puttalam

Libraries that May Be Needed:

```
java.util.HashMap;
java.util.Map;
java.util.Iterator;
java.util.Set;
```

4. Linked List

A linked list is a non-(user) indexed list of items. Again, removal of a random entry is possible.

Basic operations are:

```
add(), get(), set(), remove(), addFirst(), addLast()
removeFirst(), removeLast()
```

Declarations:

```
LinkedList<String> linkedlist  = new LinkedList<String>();
```

Iteration:

For printing:

```
System.out.println("Final Content: " +linkedlist);
```

Ex: A railway operational superintendent would like to envisage the various combinations of compartment coupling, with a linked list. Based on this, he would determine where exactly he would fit

a goods van,
an a/c compartment,
a reserved car and
a normal passenger compartment.

Create a linked list that would help him experiment with his choices.

Libraries that May Be Needed:

```
java.util.*;
```

5. Tree Map

This is much similar to the Hash Map class. Tree Map is an ordered collection however.

Basic operations are:

Similar to those in Hash Map.

Declarations:

```
TreeMap<Integer, String> tmap = new TreeMap<Integer, String>();
```

Iteration:

Similar to that in Hash Map.

Ex: Implement a tree map that would list a few of your relatives in the immediate families (with their name and relationship), according to the year they were born.

Libraries that May Be Needed:

```
java.util.TreeMap;
java.util.Map;
java.util.Iterator;
java.util.Set;
```

Q: What can be the difference between the ordering in Tree Map and Hash Map?

A: With Hash Map, it is not necessarily insertion order, but a random order every time.

6. Extended Exercise

I enter country-capital pairs, and you are supposed to write a small program that returns the country name, when a capital's name is given. For example, I enter:

"Russia" "Moscow"
"Sri Lanka" "Colombo"
"India" "New Delhi"
"Iceland" "Reykjavik".

You are not supposed to index the array by capital but you may do some overriding to fetch the data the reverse way. The program you write must return "Iceland", when you input "Reykjavik".

HINT: Search for usages of reversals of hash maps – try to do it without iteration.