Collection Framework java

Collection ??

Any group of **individual** **objects** which are represented as a single unit is known as the collection of the object

eg COLLECTION of Movies, Collection of Plants, Computers

We will collect different objects and then represent them as a single unit

Framework ?

A framework is a set of **classes** and **interfaces** which provide a ready-made architecture

Collection Framework – is a java API which provides architecture to **store** and **manipulate** group of objects

Orange = interface, green = class  
creating collection  
1) Type Safe - same type of elements(objects) are added to collection  
  
2) Un Type Safe - different types of elements can be added to collection  
  
Type Safe  
 ArrayList<String> name = **new** ArrayList<>(List.of("Satyam","Durgesh"));  
 name.add("Sachin");

Check item is there or not

Type ka equals method call karta hai chacking k lie

System.out.println(name.contains("Satyam"));  
 System.out.println(name);  
  
Un Type-Safe  
 LinkedList list = **new** LinkedList();  
 list.add("Sachin");  
 list.add(123);  
 list.add(**true**);  
 list.add(**false**);  
 System.out.println(list);

HashSet<Double> nms = **new** HashSet<>();  
nms.add(14.14);  
nms.add(23.44567);  
nms.add(123.258);  
nms.add(45.8);  
nms.add(45.8); // duplicates aren't allowed  
  
System.out.println(nms);

TreeSet<Double> treeSet = **new** TreeSet<>(nms);  
// treeSet.addAll(nms); insted of this just agrument to constructor after dimond operator  
 System.out.println(treeSet);

//traversing using Iterator  
// name.iterator(); press alt + enter to get --  
 Iterator<String> itr = name.iterator();  
  
 **while** (itr.hasNext()) {  
 String next = itr.next();  
 System.out.println(next);  
 }  
 System.out.println();  
 //backward traversal of collection ListIterator  
 ListIterator<String> litr = name.listIterator(name.size());// you need to pass size while traversing backward  
 **while** (litr.hasPrevious()) {  
 String previous = litr.previous();  
 System.out.println(previous);  
 }

For each method

name.forEach(s -> System.out.println(s));

\*\* Study comparator, comparable \*\*

HashMap

package development;

import java.util.HashMap;

public class MapEg {

public static void main(String[] args) {

HashMap<String,Integer> courses = new HashMap<>();

// adding elements

courses.put("Core java",150);

courses.put("CSA",150);

courses.put("OS",150);

// courses.put("OS",150); duplicate values are not allowed

System.out.println(courses);

courses.forEach((k,v) -> {

System.out.println(k + " => " + v); });

System.out.println(courses.get("OS"));

}}