What is Collections?

-->A collection is a single object managing a group of objects known as its elements. Types of collection -Set (HashSet, SortedSet, TreeSet) A set is an unordered collection; no duplicates are permitted in it. -List (ArrayList, LinkedList) A list is an ordered collection; duplicates are permitted. -Deque (ArrayDeque, LinkedList) -Map (HashMap, TreeMap, LinkedHashMap) A map object describes mappings from keys to values. Duplicate keys to value is not permitted but values can be duplicated.

What are Iterators?

-->Iteration is the process of retrieving every element in a collection. Various methods are hasNext(), next(), remove().

What is Comparable?

-->Comparable provides a single sorting sequence. In other

words, we can sort the collection on the basis of a single element such as id, name, and price. Comparable provides compareTo() method to sort elements. Comparable is present in java.lang package. We can sort the list elements of Comparable type by Collections.sort(List) method.

What is Comparator?

-->The Comparator provides multiple sorting sequences. In other words, we can sort the collection on the basis of multiple elements such as id, name, and price etc. Comparator provides compare() method to sort elements. A Comparator is present in the java.util package. We can sort the list elements of Comparator type by Collections.sort(List, Comparator) method.

Equals() and Hashcode() in Java:

-->The equals() and hashcode() are the two important methods provided by the Object class for comparing objects.

What is hashcode?

-->A hashcode is an integer value associated with every object in Java, facilitating the hashing in hash tables. Syntax: public int hashCode()

Returns: It returns the hash code value for the given objects.

What is equals?

-->The java equals() is a method of lang.Object class, and it is used to compare two objects. To compare two objects that whether they are the same, it compares the values of both the object's attributes. By default, two objects will be the same only if stored in the same memory location. Syntax: public boolean equals(Object obj) Parameter: obj: It takes the reference object as the parameter, with which we need to make the comparison. Returns: It returns the true if both the objects are the same, else returns false.

package Collection;

import java.util.\*;

public class Arraylist {

public static void main(String args[]){

ArrayList<String> list=new ArrayList<String>();//Creating arraylist

list.add("Rachna");

list.add("Deeksha");

list.add("Pooja");

list.add("Sandeep");

Iterator<String> itr=list.iterator();

while(itr.hasNext()){

System.out.println(itr.next());

}

}

}

package Collection;

import java.util.\*;

public class Hashmap {

public static void main(String args[]){

HashMap<Integer,String> map=new HashMap<Integer,String>();//Creating HashMap

map.put(1,"Mango"); //Put elements in Map

map.put(2,"Apple");

map.put(3,"Banana");

map.put(4,"Grapes");

System.out.println("Iterating Hashmap...");

for(Map.Entry m : map.entrySet()){

System.out.println(m.getKey()+" "+m.getValue());

}

}

}

package Collection;

import java.util.\*;

public class Hashset {

public static void main(String[] args)

{

HashSet<String> hs= new HashSet<String>();

hs.add("India");

hs.add("America");

hs.add("Russia");

hs.add("China");

hs.add("India"); //duplicate value

hs.add("Russia"); //duplicate value

System.out.println("Set is "+hs); //view HashSet

Iterator<String> it=hs.iterator(); //add an iterator to hs

System.out.println("Elements using iterator:");

while(it.hasNext()) //display elements by using iterator

{

String s=(String)it.next();

System.out.println(s);

}

}

}