**Adv. Java Lab Assignment**

**Collections 20 Programs**

**Name-Milind Bajaj**

**M.Tech(5th Sem)**

**Roll No.:- IT-2k17-35**

**1.Program of adding elements in ArrayList .**

**Solution:-**

import java.util.\*;

class Student{

int rollno;

String name;

int age;

Student(int rollno,String name,int age){

this.rollno=rollno;

this.name=name;

this.age=age;

}

}

class ArrayListClass{

public static void main(String args[]){

Student s1=new Student(101,"thor",23);

Student s2=new Student(102,"marvel",21);

Student s3=new Student(103,"loki",25);

ArrayList<Student> al=new ArrayList<Student>();

al.add(s1);

al.add(s2);

al.add(s3);

Iterator itr=al.iterator;

while(itr.hasNext()){

Student st=(Student)itr.next();

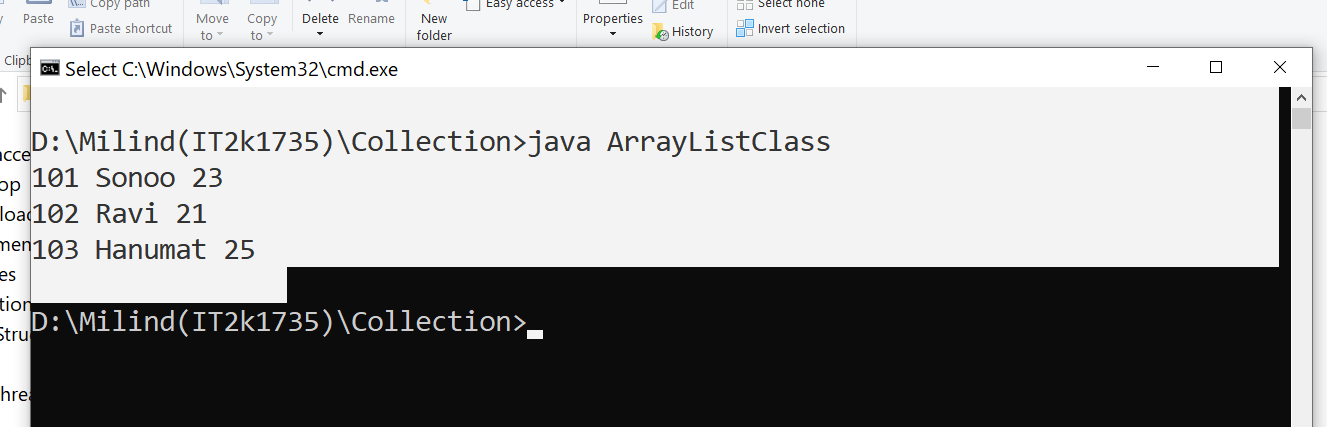
System.out.println(st.rollno+" "+st.name+" "+st.age);

}

}

}

**Output 1:-**

****

**2.Program of ArrayList (getting elements with all methods).**

**Solution:-**

import java.util.\*;

class ArrayListAll{

public static void main(String args[]){

ArrayList<String> list=new ArrayList<String>();

list.add("Thor");

list.add("Ironman");

list.add("hulk");

list.add("marvel”);

System.out.println("Traversing list through List Iterator:");

ListIterator<String> list1=list.listIterator(list.size());

while(list1.hasPrevious())

{

String str=list1.previous();

System.out.println(str);

}

System.out.println("Traversing list through for loop:");

for(int i=0;i<list.size();i++)

{

System.out.println(list.get(i));

}

System.out.println("Traversing list through forEach() method:");

list.forEach(a->System.out.println(a);

});

System.out.println("Traversing list through forEachRemaining() method:");

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

itr.forEachRemaining(a-> (

{

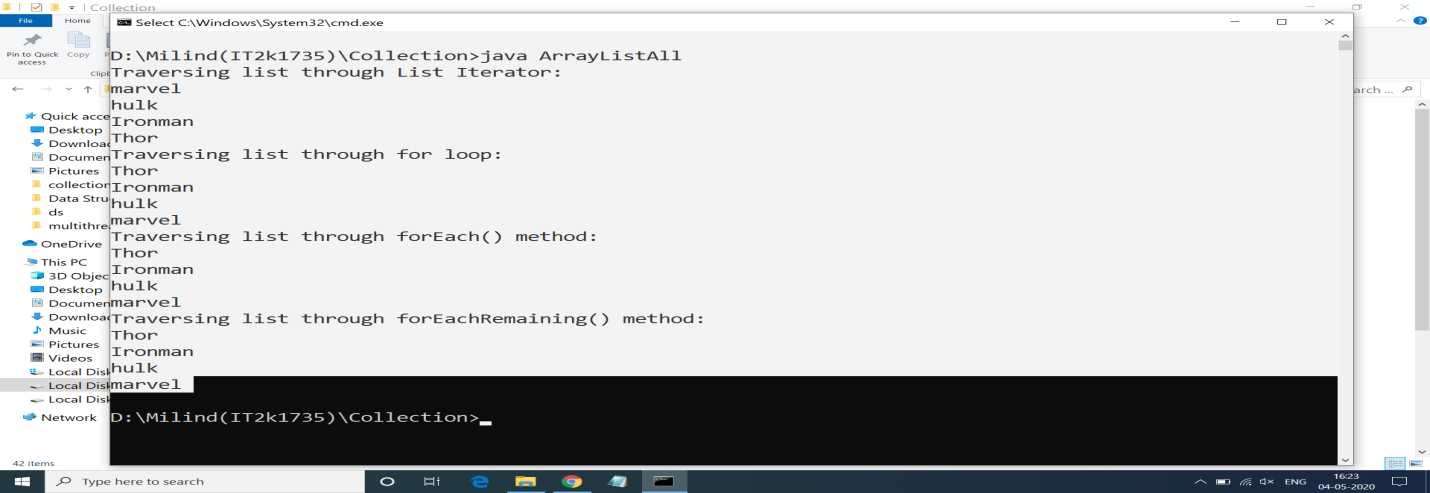
System.out.println(a);

});

}

}

**Output 2:-**

****

**3.Comapring Elements with compareTo() Method.**

**Solution:-**

import java.util.\*;

public class compareto {

public static void main(String[] args) {

String s1 = "Hello";

String s2 = "World!";

String s3 = "";

System.out.println("s1 compareto s2: " + s1.compareTo(s2));

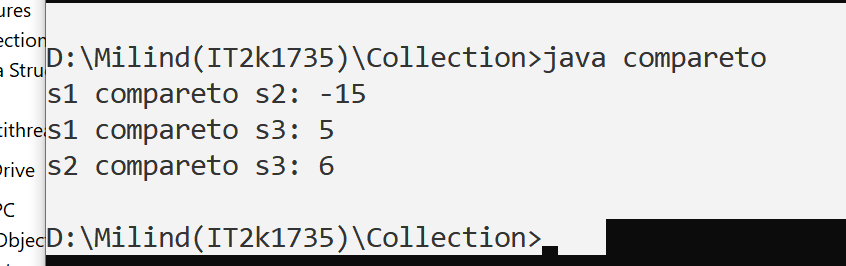
System.out.println("s1 compareto s3: " + s1.compareTo(s3));

System.out.println("s2 compareto s3: " + s2.compareTo(s3));

}

}

**Output 3:-**

****

**4.Program of Enumeration in Collection.**

**Solution:-**

public class Enum {

public enum Season {WINTER, SPRING, SUMMER, FALL}

public static void main(String[] args)

{

System.out.println("Getting season with enum:");

for (Season s : Season.values()) {

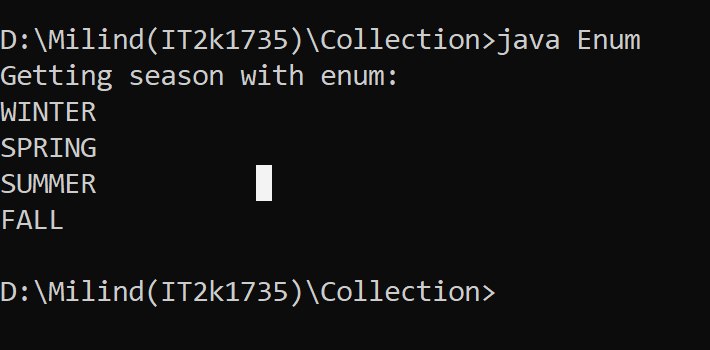
System.out.println(s);

}

}

}

**Output 4:-**

****

**5.Program of making LinkedList using Collection.**

**Solution:-**

import java.util.Iterator;

import java.util.LinkedList;

import java.util.List;

public class Linkedlst {

public static void main(String[] args) {

List<String> lst = new LinkedList<String>();

lst.add("Thor");

lst.add("hulk");

lst.add("Ironman");

lst.add("Marvel");

Iterator itr = lst.iterator();

Object First = itr.next();

Object Second = itr.next();

Object third = itr.next();

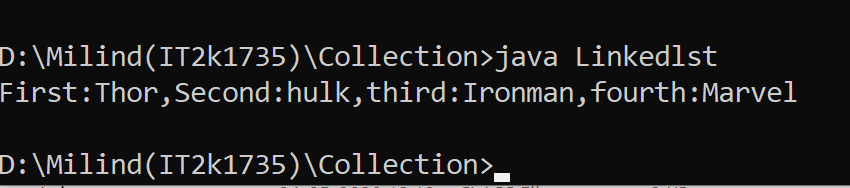
Object fourth = itr.next();

System.out.println("First:"+First+",Second:"+Second+",third:"+third+",fourth:"+fourth);

}

}

**Output 5:-**

****

**6.Program of Removing elements from linkedlist.**

**Solution:-**

import java.util.LinkedList;

public class Linkedlst1 {

public static void main(String[] args) {

LinkedList<String> lst = new LinkedList<String>();

lst.add("xmen");

lst.add("hulk");

lst.add("thor");

lst.add("ironman");

System.out.println(lst);

System.out.println("remove :" + lst.remove(0));

System.out.println(lst);

System.out.println("remove:" + lst.remove(1));

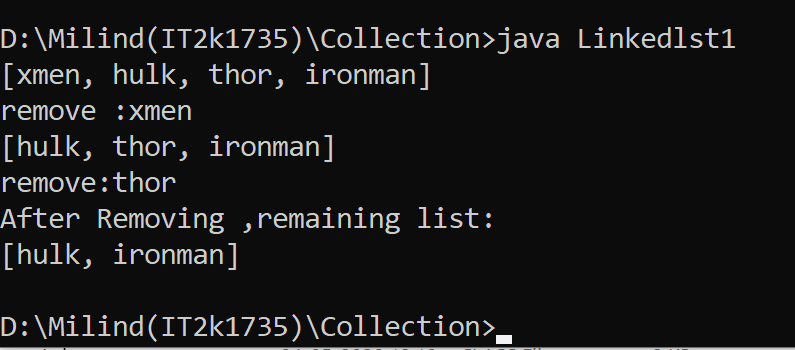
System.out.println("After Removing ,remaining list:");

System.out.println(lst);

}

}

**Output 6:-**

****

**7.Program of making Queue using Collection.**

**Solution:-**

import java.util.LinkedList;

import java.util.Queue;

public class Queue1 {

public static void main(String[] args) {

Queue<Integer> qu = new LinkedList<>();

for (int i = 0; i < 5; i++) {

qu.add(i);

}

System.out.println("Queue Element:" + qu);

Integer remo = qu.remove();

System.out.println("removed Element:" + remo);

Integer head = qu.peek();

System.out.println("Head of queue:" + head);

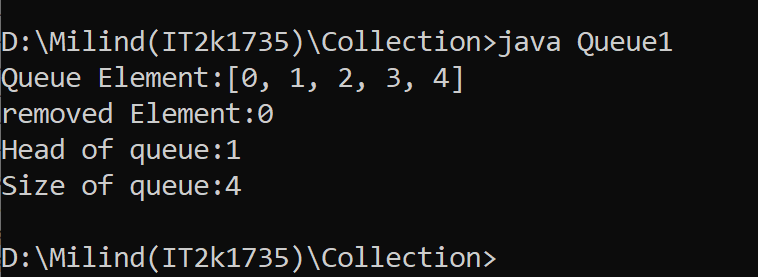
int size = qu.size();

System.out.println("Size of queue:" + size);

}

}

**Output 7:-**

****

**8.Program of creating Priority Queue using Collection.**

**Solution:-**

import java.util.Iterator;

import java.util.PriorityQueue;

class Pqueue {

public static void main(String[] args) {

PriorityQueue<String> q = new PriorityQueue<>();

q.add("navlakha");

q.add("palasiya");

q.add("56 dukan");

System.out.println("head" + q.peek());

System.out.println("head:" + q.element());

System.out.println("iterating the queue Element");

Iterator itr = q.iterator();

while (itr.hasNext()) {

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

}

System.out.println("after removing two Element");

q.remove();

q.poll();

Iterator itr1 = q.iterator();

while (itr1.hasNext()) {

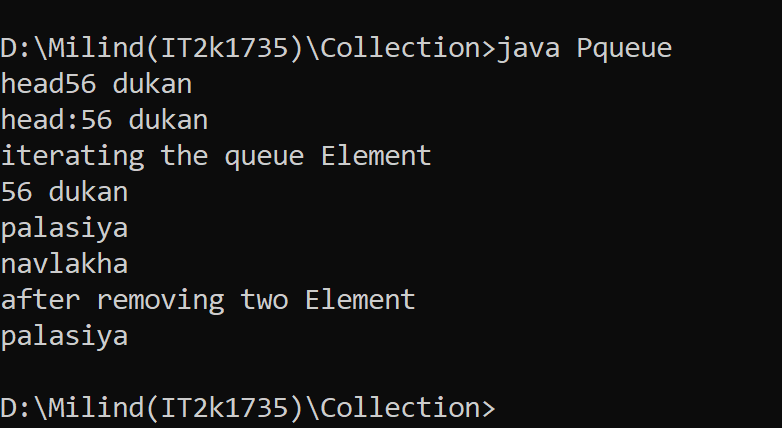
System.out.println(itr1.next());

}

}

}

**Output 8:-**

****

**9.Program of creating Stack using Collection.**

**Solution:-**

import java.io.\*;

import java.util.\*;

class StackDemo

{

static void stack\_push(Stack<Integer> stack)

{

for(int i = 0; i < 5; i++)

{

stack.push(i);

}

}

static void stack\_pop(Stack<Integer> stack)

{

System.out.println("Pop :");

for(int i = 0; i < 5; i++)

{

Integer y = (Integer) stack.pop();

System.out.println(y);

}

}

static void stack\_peek(Stack<Integer> stack)

{

Integer element = (Integer) stack.peek();

System.out.println("Element on stack top : " + element);

}

static void stack\_search(Stack<Integer> stack, int element)

{

Integer pos = (Integer) stack.search(element);

if(pos == -1)

System.out.println("Element not found");

else

System.out.println("Element is found at position " + pos);

}

public static void main (String[] args)

{

Stack<Integer> stack = new Stack<Integer>();

stack\_push(stack);

stack\_pop(stack);

stack\_push(stack);

stack\_peek(stack);

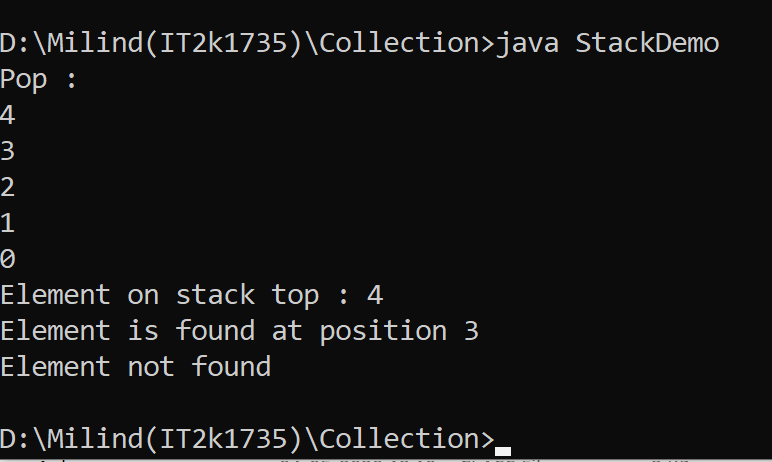
stack\_search(stack, 2);

stack\_search(stack, 6);

}

}

**Output 9:-**

****

**10.Program of Map in Collection interface.**

**Solution:-**

import java.util.HashMap;

import java.util.Iterator;

import java.util.Map;

import java.util.Set;

public class MyMap {

public static void main(String[] args) {

Map m = new HashMap();

m.put(1, "dhoni");

m.put(7, "Rahul");

m.put(9, "ram");

m.put(6, "pant");

Set set = m.entrySet();

Iterator itr = set.iterator();

while (itr.hasNext()) {

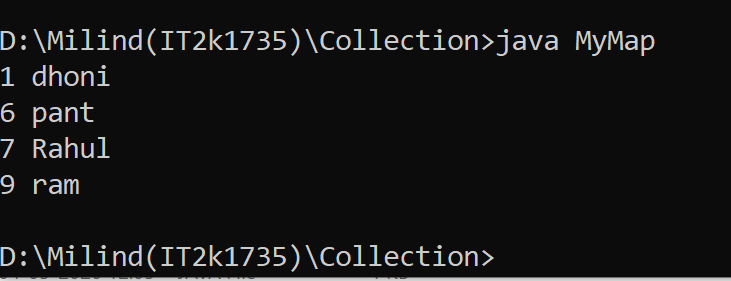
Map.Entry entry = (Map.Entry) itr.next();

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

}

}}

**Output 10:-**



**11.Program of Sorted Map with Collection.**

**Solution:-**

import java.util.SortedMap;

import java.util.TreeMap;

public class SortedMap1 {

public static void main(String[] args) {

SortedMap<Integer, String> sm = new TreeMap<Integer, String>();

sm.put(107, "Dhoni");

sm.put(102, "Rohit Sharma");

sm.put(106, "Virat");

System.out.println("HeadMap:" + sm.headMap(107));

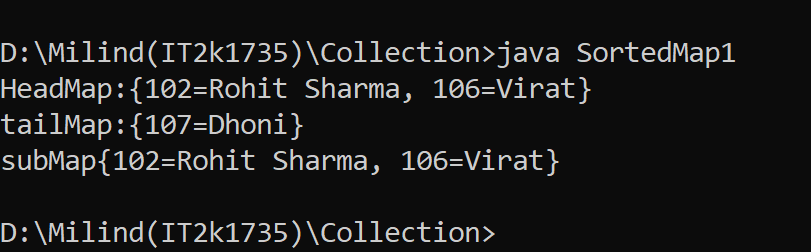
System.out.println("tailMap:" + sm.tailMap(107));

System.out.println("subMap" + sm.subMap(102, 107));

}

}

**Output 11:-**

****

**12.Program Of Identity Hash Map.**

**Solution:-**

import java.util.HashMap;

import java.util.IdentityHashMap;

import java.util.Map;

public class Identityhashmap {

public static void main(String[] args) {

Map h = new HashMap();

Map hm = new IdentityHashMap();

h.put("hkey", "hvalue");

hm.put("hmkey", "hmvalue");

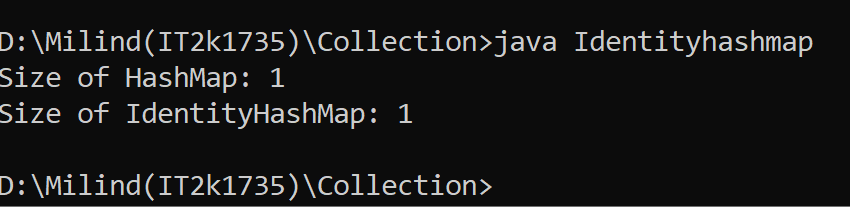
System.out.println("Size of HashMap: " + h.size());

System.out.println("Size of IdentityHashMap: " + hm.size());

}

}

**Output 12:-**

****

**13.Program of Hash Set in Collection.**

**Solution:-**

import java.util.HashSet;

public class Hashset1 {

public static void main(String[] args) {

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

hs.add("Virat kohli");

hs.add("Rohit sharma");

hs.add("Rishabh pant");

hs.add("Ms Dhoni");

System.out.println(hs);

System.out.println("List contains Ms Dhoni or not :" + hs.contains("Ms Dhoni"));

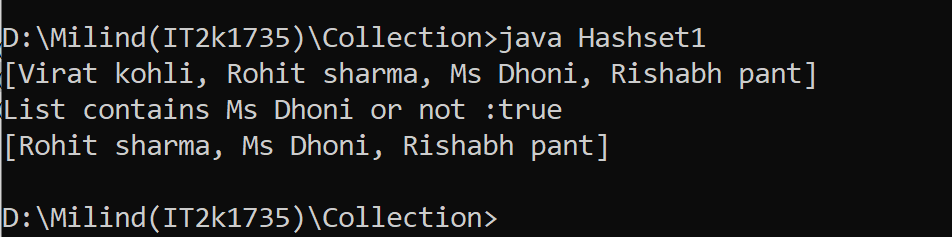
hs.remove("Virat kohli");

System.out.println(hs);

}

}

**Output 13:-**

****

**14.Program of LinkedHash Set in Collection.**

**Solution:-**

import java.util.LinkedHashSet;

public class Linkedhashset {

public static void main(String[] args) {

LinkedHashSet<String> a = new LinkedHashSet<>();

a.add("kalu");

a.add("ram");

a.add("lakhan");

System.out.println("linkedhashset size:" + a.size());

System.out.println("Original Linkedhashset:" + a);

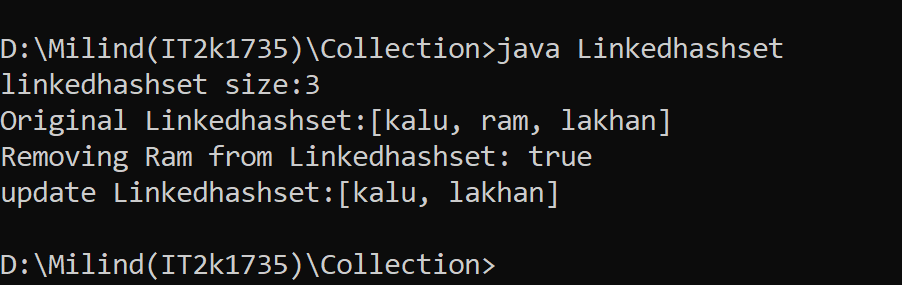
System.out.println("Removing Ram from Linkedhashset: " + a.remove("ram"));

System.out.println("update Linkedhashset:" + a);

}

}

**Output 14:-**

****

**15.Program of Tree in Collection.**

**Solution:-**

import java.util.SortedSet;

import java.util.TreeSet;

public class Tree1 {

public static void main(String[] args) {

SortedSet<String> str = new TreeSet<>();

str.add("mumbai");

str.add("delhi");

str.add("chennai");

str.add("kolkata");

str.add("hyderabad");

System.out.println("getting elements in tree");

for (String s : str) {

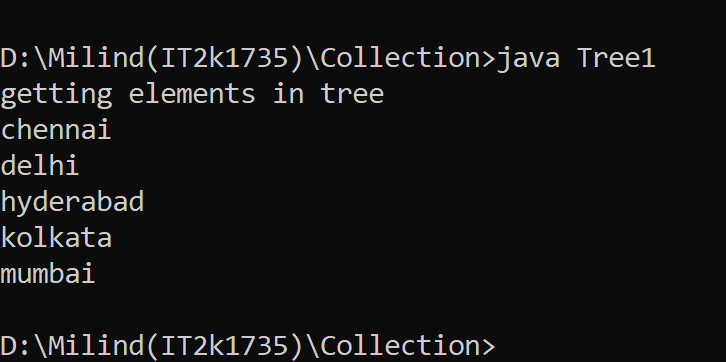
System.out.println(s);

}

}

}

**Output 15:-**

****

**16. Program of TreeMap in Collection Framework.**

**Solution:-**

import java.util.Map;

import java.util.TreeMap;

public class TreeMap1 {

public static void main(String[] args) {

TreeMap<Integer, String> tr = new TreeMap<>();

tr.put(1, "Virat kohli");

tr.put(2, "Pant");

tr.put(3, "Dhoni");

tr.put(4, "Rohit");

for (Map.Entry a : tr.entrySet()) {

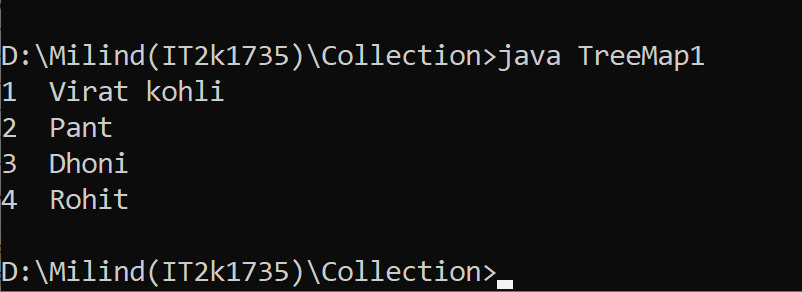
System.out.println(a.getKey() + " " + a.getValue());

}

}

}

**Output 16:-**

****

**17.Program of TreeSet with Collection.**

**Solution:-**

import java.util.TreeSet;

public class TreeSet1 {

public static void main(String[] args) {

TreeSet<String> set = new TreeSet<String>();

set.add("A");

set.add("B");

set.add("C");

set.add("D");

set.add("E");

System.out.println("Initial Set: " + set);

System.out.println("Reverse Set: " + set.descendingSet());

System.out.println("Head Set: " + set.headSet("C", true));

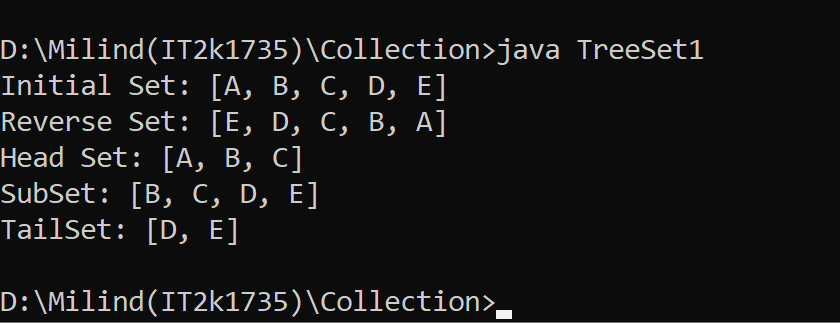
System.out.println("SubSet: " + set.subSet("A", false, "E", true));

System.out.println("TailSet: " + set.tailSet("C", false));

}

}

**Output 17:-**

****

**18.Program of Vector in Collection.**

**Solution:-**

import java.util.Iterator;

import java.util.Vector;

public class Vector1 {

public static void main(String[] args) {

Vector<String> a = new Vector<>();

a.add("modi");

a.add("trump");

a.add("imran");

Iterator itr = a.iterator();

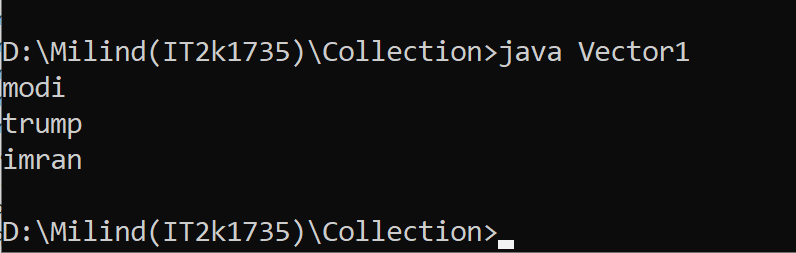
while (itr.hasNext()) {

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

}

}}

**Output 18:-**

****

**19.Program of WeakHashMap in Collection.**

**Solution:-**

import java.util.HashMap;

import java.util.Map;

import java.util.Set;

public class WeakhashMap1 {

public static void main(String[] args) {

Map<Number, String> a = new HashMap<>();

a.put(1, "thor");

a.put(2, "hulk");

a.put(3, "iornman");

Set set = a.entrySet();

System.out.println(set);

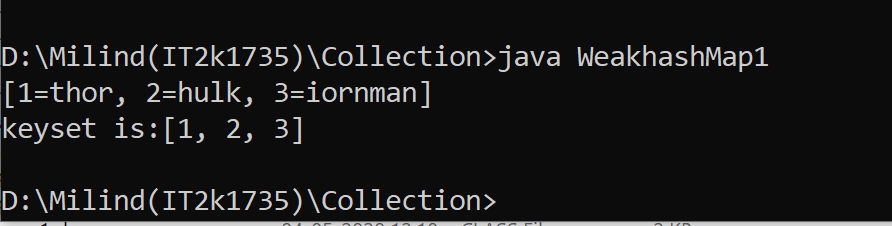
Set keySet = a.keySet();

System.out.println("keyset is:" + keySet);

}

}

**Output 19:-**

****

**20.Program of WeakHashMap in Collection.**

**Solution:-**

import java.util.HashMap;

import java.util.Map;

public class WeakhashMap2 {

public static void main(String[] args) {

Map<Number, String> a = new HashMap<>();

a.put(1, "marvel");

a.put(2, "antman");

a.put(3, "nat");

Map<Number, String> a1 = new HashMap<>();

a1.putAll(a);

System.out.println("getting second element:"+a1.get(2));

System.out.println("Size of map is:" + a1.size());

System.out.println("removing second element:");

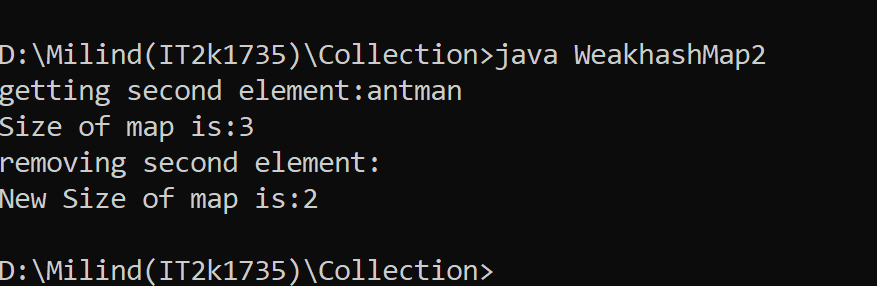
a1.remove(2);

System.out.println("New Size of map is:" + a1.size());

}

}

**Output 20:-**

****

**END**