**Assignments on Collections**

1) Given a TreeMap<Long, Contact> which has phone numbers for keys and contact objects for values.

Write solutions to

a. Fetch all the keys and print them,

b. Fetch all the values and print them

c. Print all key-value pairs

Note:

a) Contacts should be stored in descending order of phone number

b) Contact Class:

. PhoneNumer: <long>

. Name: <String>

Email: <String>

**package** map;

**import** java.util.\*;

**public** **class** Contact{

Long phoneNum;

String name , gmail;

String gen;

Contact( String name , String gmail, String gen ){

**this**.name=name;

**this**.gmail=gmail;

**this**.gen=gen;

}

**public** **void** setphoneNo(Long no) {

**this**.phoneNum=no;

}

**public** Long getphoneNo() {

**return** phoneNum;

}

**public** **void** setname(String name) {

**this**.name=name;

}

**public** String getname() {

**return** name;

}

**public** **void** setgmail(String gmail) {

**this**.gmail=gmail;

}

**public** String getgmail() {

**return** gmail;

}

**public** String toString()

{

**return** (name+"--"+gmail+"--"+gen);

}

}

**package** map;

**import** java.util.\*;

**public** **class** TreeMapAssign {

**private** **static** **final** Object ***Contact*** = **null**;

**public** **static** **void** main(String[] args) {

TreeMap t=**new** TreeMap();

Contact c1=**new** Contact ("pranjal", "pranjaldongre123" , "male");

Contact c2=**new** Contact("shubh", "shubh123" , "male");

Contact c3=**new** Contact("abhishek", "abhiabhi153" , "male");

Contact c4=**new** Contact("ravina", "ravina1223" , "female");

t.put(882543365,c1);

t.put(739463822,c2);

t.put(363929735,c3);

t.put(548393654,c4);

System.***out***.println(t);

}

}

2) Write an application to store 10 unique product objects. In case there is an attempt to add a duplicate product, it should be silently rejected. Hint: Use HasSet or TreeSet

**package** map;

**import** java.util.\*;

**public** **class** abc **implements** Comparator<Contact>{

@Override

**public** **int** compare(Contact c1, Contact c2) {

**return** (-c1.getphoneNo().compareTo(c2.getphoneNo()));

}

}

**package** assignColl;

**import** java.util.\*;

**public** **class** TreeeSet {

**public** **static** **void** main(String[] args) {

TreeSet t=**new** TreeSet();

t.add("apple");

t.add("dell");

t.add("hp");

t.add("vivo");

t.add("micromax");

t.add("apple");

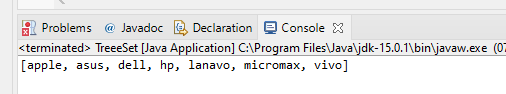
t.add("lanavo");

t.add("asus");

System.***out***.println(t);

}

}



**package** ES;

**import** java.util.Comparator;

**class** Employee{

**int** id;

String name ;

String department;

**double** salary;

**public** Employee(**int** id, String name, String department, **double** salary)

{

**this**.id=id;

**this**.name=name;

**this**.department= department;

**this**.salary= salary;

}

**public** **int** getId() {

**return** id;

}

**public** **void** setId(**int** id) {

**this**.id = id;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

**public** String getDepartment() {

**return** department;

}

**public** **void** setDepartment(String department) {

**this**.department = department;

}

**public** **double** getSalary() {

**return** salary;

}

**public** **void** setSalary(**double** salary) {

**this**.salary = salary;

}

**public** String toString() {

**return** (id+"---"+name+"---"+department+"---"+salary);

}

}

**package** ES;

**import** java.util.Comparator;

**class** myComparator **implements** Comparator<Employee>

{

@Override

**public** **int** compare(Employee o1, Employee o2) {

**return** o1.getName().compareTo(o2.getName());

//return (int) (o1.getSalary() - o2.getSalary());

}

}

**package** ES;

**import** java.util.\*;

**public** **class** user {

**public** **static** **void** main(String[] args) {

TreeSet<Employee>t=**new** TreeSet<Employee>(**new** myComparator());

Employee e1=**new** Employee(1234 , "Pranjal" , "Security" , 8000);

Employee e2=**new** Employee(5678, "Ravi" , "IT" , 81000);

Employee e3=**new** Employee(9364 , "kishan" , "Advertisement" , 7000);

Employee e4=**new** Employee(3526, "shubham" , "News" , 2000);

Employee e5=**new** Employee(6256 , "Krishna" , "customer" , 9000);

Employee e6=**new** Employee(2532 , "Keshav" , "Social Media" , 3000);

Employee e7=**new** Employee(4273 , "Kiran" , "Marketing" , 6000);

Employee e8=**new** Employee(5835 , "Abhishek" , "salery" , 4000);

t.add(e1);

t.add(e2);

t.add(e3);

t.add(e4);

t.add(e5);

t.add(e6);

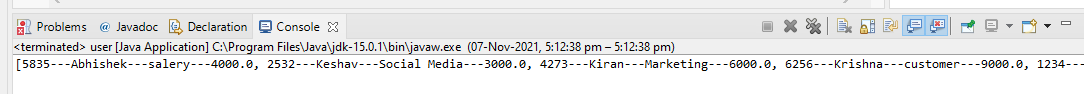
t.add(e7);

t.add(e8);

System.***out***.println(t);

}

}



**package** assignColl;

**import** java.text.ParseException;

**import** java.time.LocalDate;

**import** java.time.format.DateTimeFormatter;

**import** java.util.LinkedList;

;

**public** **class** Birthdate {

**public** **static** **void** main(String[] args ) **throws** ParseException

{

LinkedList<LocalDate> ll= **new** LinkedList<LocalDate>();

ll.add(LocalDate.of(2016, 07,23));

ll.add(LocalDate.of (1999,02,01));

ll.add(LocalDate.of(2000, 12,05));

ll.add(LocalDate.of(2019,10,24));

ll.add(LocalDate.of (1974,07,30));

ll.add(LocalDate.of (1996,11,18));

**while**(itr.hasNext())

LocalDate d=itr.next();

DateTimeFormatter formatter = DateTimeFormatter.ofPattern("dd-MM-yyyy");

String fd = d.format(formatter);

**if** (d.isLeapYear()) {

System.out.println("Your date of birth is " + fd + " and it was a leap year.");

}

**else**

{

System.out.println("Your date of birth is "+fd +" and it was not a leap year.");

}