CORE JAVA

COLLECTION FRAMEWORK ASSIGNMENT

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:
      - PhoneNumber:<long>
      - Name:<String>
      - Email:<String>
      - Gender:<Enum>

   Code: -

ContactTree.java × HashSetExample.java EmployeeObject.java CheckLeapYear.java

```
1  package collection;
2  import java.util.TreeMap;
3  class Contact
4  {
5  long PhoneNumber;
6  String Name;
7  String Email;
8  String Gender;
9  public Contact(long phoneNumber, String name, String email, String gender)
10 {
11 super();
12 PhoneNumber = phoneNumber;
13 Name = name;
14 Email = email;
15 Gender = gender;
16 }
17 @Override
18 public String toString()
```

Problems Javadoc Declaration Console × Git Staging

```
<terminated> ContactTree [Java Application] C:\Users\MBALKRIS\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.0.v20211012-105
Fetching all the keys
877991234
8108764545
8655454545
Fetching all the Values
[Number=8876543901, Name=Siddhi, Email=sid@gmail.com, Gender=Female]

[Number=9930697398, Name=Mrunal, Email=mru@gmail.com, Gender=Female]

[Number=9967235410, Name=Kavish, Email=kav@gmail.com, Gender=Male]

Printing all the Key-Values pairs:{877991234=[Number=8876543901, Name=Siddhi, Email=sid@gmail.com, Gender=Female]
, 8108764545=[Number=9930697398, Name=Mrunal, Email=mru@gmail.com, Gender=Female]
, 8655454545=[Number=9967235410, Name=Kavish, Email=kav@gmail.com, Gender=Male]
}
```

```
ContactTree.java × HashSetExample.java  EmployeeObject.java  CheckLeapYear.java
16 }
17@Override
18 public String toString()
19 {
20 return "[Number=" + PhoneNumber + ", Name=" + Name + ", Email=" + Email + ", Gender=" + Gender + "]" + "\n";
21 }
22 }
23 public class ContactTree
24 {
25public static void main(String[] args)
26 {
27 Contact obj1 = new Contact( 9930697398L, "Mrunal" , "mru@gmail.com" , "Female");
28 Contact obj2 = new Contact( 8876543901L, "Siddhi" , "sid@gmail.com" , "Female");
29 Contact obj3 = new Contact( 9967235410L, "Kavish" , "kav@gmail.com" , "Male");
30 TreeMap < Long , Contact> tr = new TreeMap<Long , Contact>();
31 tr.put(8108764545L, obj1);
32 tr.put(877991234L, obj2);
33 tr.put(8655454545L, obj3);
34 System.out.println("Fetching all the keys");
35 for(Long intk : tr.keySet())
36 {
37 System.out.println(intk);
38 }
39 System.out.println("Fetching all the Values");
40 for (Contact strV : tr.values())
41 {
42 System.out.println(strV);
43 }
44 System.out.println("Printing all the Key-Values pairs:"+ tr);
45 }
46 }
47
```

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 HashSet or TreeSet Extra(optional): Use ArrayList in the above solution. (This is optional)

   Code: -

```
ContactTree.java  HashSetExample.java ×  EmployeeObject.java  Check
 1 package collection;
 2 import java.util.HashSet;
 3 public class HashSetExample
 4 {
 5public static void main(String[] args)
 6 {
 7
 8              HashSet<Integer> myhashset = new HashSet<>();
 9              myhashset.add(11);
10              myhashset.add(21);
11              myhashset.add(3);
12              myhashset.add(4);
13              myhashset.add(50);
14              myhashset.add(6);
15              myhashset.add(7);
16              myhashset.add(87);
17              myhashset.add(9);
18              myhashset.add(10);
19              myhashset.add(10);
20              myhashset.add(11);
21              System.out.println(myhashset);
22
23     }
24 }
```

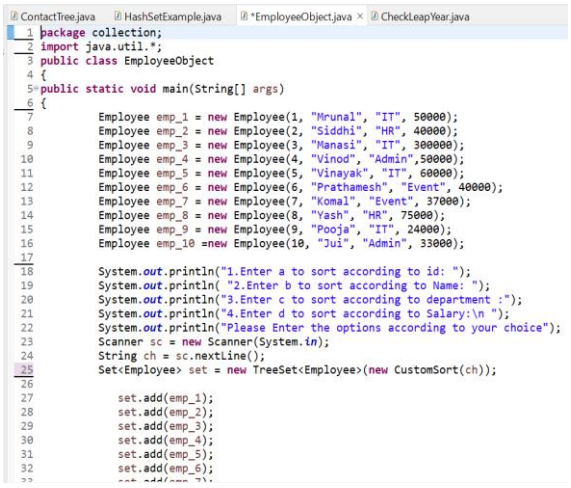
```
Problems  Javadoc  Declaration  Console ×  Git Staging
<terminated> HashSetExample [Java Application] C:\Users\MBALKRIS\.p2\pool\plugir
[50, 3, 4, 21, 6, 7, 87, 9, 10, 11]
```

3. Store at least 10 Employee Objects in an TreeSet<Employee>. When the application runs the user should be asked to select one of the options upon which you will print the employee details in a sorted manner.
   For e.g.
   Run Application:
   a) ID
   b) Name
   c) Department
   d) Salary

   Your choice: b

   <Should print all the employees details sorted by name>

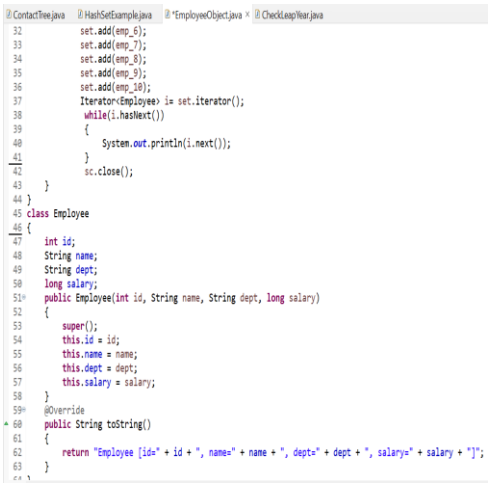
   Code: -



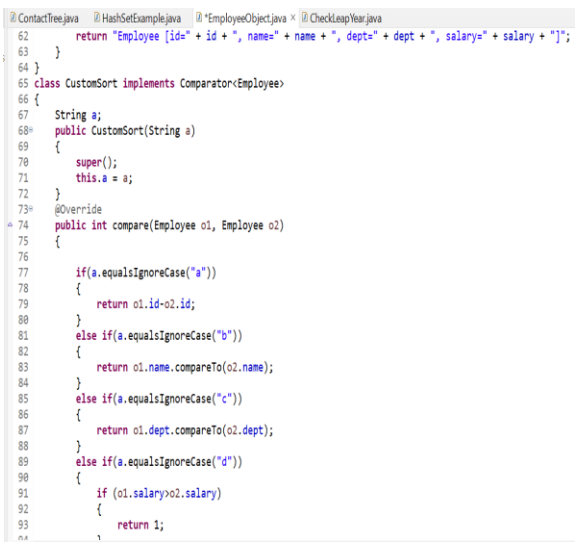
```
package collection;
import java.util.*;
public class EmployeeObject
{
public static void main(String[] args)
{
        Employee emp_1 = new Employee(1, "Mrunal", "IT", 50000);
        Employee emp_2 = new Employee(2, "Siddhi", "HR", 40000);
        Employee emp_3 = new Employee(3, "Manasi", "IT", 300000);
        Employee emp_4 = new Employee(4, "Vinod", "Admin",50000);
        Employee emp_5 = new Employee(5, "VinayaK", "IT", 60000);
        Employee emp_6 = new Employee(6, "Prathamesh", "Event", 40000);
        Employee emp_7 = new Employee(7, "Komal", "Event", 37000);
        Employee emp_8 = new Employee(8, "Yash", "HR", 75000);
        Employee emp_9 = new Employee(9, "Pooja", "IT", 24000);
        Employee emp_10 =new Employee(10, "Jui", "Admin", 33000);

        System.out.println("1.Enter a to sort according to id: ");
        System.out.println( "2.Enter b to sort according to Name: ");
        System.out.println("3.Enter c to sort according to department :");
        System.out.println("4.Enter d to sort according to Salary:\n ");
        System.out.println("Please Enter the options according to your choice");
        Scanner sc = new Scanner(System.in);
        String ch = sc.nextLine();
        Set<Employee> set = new TreeSet<Employee>(new CustomSort(ch));

           set.add(emp_1);
           set.add(emp_2);
           set.add(emp_3);
           set.add(emp_4);
           set.add(emp_5);
           set.add(emp_6);
```

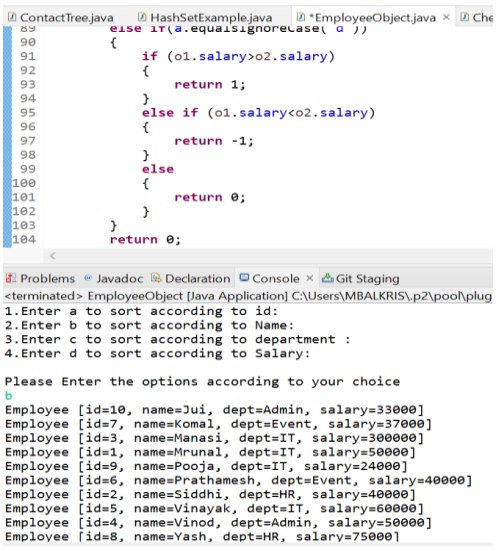


```
        set.add(emp_6);
        set.add(emp_7);
        set.add(emp_8);
        set.add(emp_9);
        set.add(emp_10);
        Iterator<Employee> i= set.iterator();
         while(i.hasNext())
         {
             System.out.println(i.next());
         }
         sc.close();
    }
}
class Employee
{
    int id;
    String name;
    String dept;
    long salary;
    public Employee(int id, String name, String dept, long salary)
    {
        super();
        this.id = id;
        this.name = name;
        this.dept = dept;
        this.salary = salary;
    }
    @Override
    public String toString()
    {
        return "Employee [id=" + id + ", name=" + name + ", dept=" + dept + ", salary=" + salary + "]";
    }
```



```
        return "Employee [id=" + id + ", name=" + name + ", dept=" + dept + ", salary=" + salary + "]";
    }
}
class CustomSort implements Comparator<Employee>
{
    String a;
    public CustomSort(String a)
    {
        super();
        this.a = a;
    }
    @Override
    public int compare(Employee o1, Employee o2)
    {

        if(a.equalsIgnoreCase("a"))
        {
            return o1.id-o2.id;
        }
        else if(a.equalsIgnoreCase("b"))
        {
            return o1.name.compareTo(o2.name);
        }
        else if(a.equalsIgnoreCase("c"))
        {
            return o1.dept.compareTo(o2.dept);
        }
        else if(a.equalsIgnoreCase("d"))
        {
            if (o1.salary>o2.salary)
            {
                return 1;
```



```
        else if(a.equalsIgnoreCase("d"))
        {
            if (o1.salary>o2.salary)
            {
                return 1;
            }
            else if (o1.salary<o2.salary)
            {
                return -1;
            }
            else
            {
                return 0;
            }
        }
        return 0;
```

```
Problems  Javadoc  Declaration  Console  Git Staging
<terminated> EmployeeObject [Java Application] C:\Users\MBALKRIS\.p2\pool\plug
1.Enter a to sort according to id:
2.Enter b to sort according to Name:
3.Enter c to sort according to department :
4.Enter d to sort according to Salary:

Please Enter the options according to your choice
b
Employee [id=10, name=Jui, dept=Admin, salary=33000]
Employee [id=7, name=Komal, dept=Event, salary=37000]
Employee [id=3, name=Manasi, dept=IT, salary=300000]
Employee [id=1, name=Mrunal, dept=IT, salary=50000]
Employee [id=9, name=Pooja, dept=IT, salary=24000]
Employee [id=6, name=Prathamesh, dept=Event, salary=40000]
Employee [id=2, name=Siddhi, dept=HR, salary=40000]
Employee [id=5, name=Vinayak, dept=IT, salary=60000]
Employee [id=4, name=Vinod, dept=Admin, salary=50000]
Employee [id=8, name=Yash, dept=HR, salary=75000]
```

4. Given a LinkedList of Objects representing date of birth's (use any inbuild java class to represent date), print the date's along with the message: Your date of Birth is DD-MM-YYYY, and it (was or was not) a leap year.
   E.g.
   a) For the date 23-12-2000

Your date of Birth is 23-12-2000 and it was a leap year

b) For the date 23-12-2001

Your date of Birth is 23-12-2001 and it was not a leap year

Note: - You need to access the Dates in the reverse order. i.e. start from the last object and move towards the first object.

Code: -

ContactTree.java HashSetExample.java EmployeeObject.java

```
1  package collection;
2  import java.time.LocalDate;
3  import java.time.format.DateTimeFormatter;
4  import java.util.LinkedList;
5  import java.util.List;
6
7  public class CheckLeapYear
8  {
9  public static void main(String[] args)
10 {
11         Date date = new Date("22/08/1998");
12         Date date1 = new Date("12/10/1995");
13         Date date2 = new Date("13/04/2017");
14         Date date3 = new Date("15/09/2012");
15         Date date4 = new Date("12/10/2016");
16         Date date5= new Date("26/08/2004");
17         Date date6 = new Date("10/12/2005");
18
19         List<Date> dobList = new LinkedList<>();
20         dobList.add(date);
21         dobList.add(date1);
22         dobList.add(date2);
23         dobList.add(date3);
24         dobList.add(date4);
25         dobList.add(date5);
```

Problems Javadoc Declaration Console Git Staging

<terminated> CheckLeapYear [Java Application] C:\Users\MBALKRIS\.p2\p

```
22/08/1998 This is not a leap year
12/10/1995 This is not a leap year
13/04/2017 This is not a leap year
15/09/2012 This is a leap year
12/10/2016 This is a leap year
26/08/2004 This is a leap year
10/12/2005 This is not a leap year
```

ContactTree.java HashSetExample.java EmployeeObject.java CheckLeapYear.java

```
26         dobList.add(date6);
27
28         DateTimeFormatter df = DateTimeFormatter.ofPattern("dd/MM/yyyy");
29
30         for(int i =0;i<dobList.size();i++)
31         {
32             LocalDate ld =LocalDate.parse(dobList.get(i).date,df);
33             String sd = (ld).format(df);
34
35             if (ld.getYear()%4 ==0)
36             {
37                 System.out.println(sd + " "+"This is a leap year");
38             }
39             else
40             {
41                 System.out.println(sd+" "+"This is not a leap year");
42             }
43         }
44 }
45 }
46 class Date
47 {
48     String date;
49     public Date(String date)
50     {
51         super();
52         this.date = date;
53     }
54     @Override
55     public String toString()
56     {
57         return " [date=" + date + "]";
58     }
```

```
46 class Date
47 {
48     String date;
49     public Date(String date)
50     {
51         super();
52         this.date = date;
53     }
54     @Override
55     public String toString()
56     {
57         return " [date=" + date + "]";
58     }
59     public String getDate()
60     {
61         return date;
62     }
63 }
64
```