COLLECTION ASSIGNMENT

1)Contact:

```
public class Contact{
    String name;
    String email;
    Enum Gender;

    public Contact(String name,String email,Enum gender) {
        this.name=name;
        this.email=email;
        this.Gender=gender;
    }



    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    public String getEmail() {
        return email;
    }

    public void setEmail(String email) {
        this.email = email;
    }


    public Enum getGender() {
        return Gender;
    }

    public void setGender(Enum gender) {
        Gender = gender;
     }

     public String toString()
     {
         return "Contact [name="+name+",email="+email+",gender="+Gender+"]";
     }
}
```

```
import java.util.Collections;

class Main {

    enum gender{
        Female, Male
    }

    public static void main(String[] args) {

        Main.gender gender = null;
        gender f1=gender.Female;
        gender m1=gender.Male;


        TreeMap<Long,Contact> contact = new TreeMap<Long,Contact>(Collections.reverseOrder());

                Contact Shruti = new Contact("Shruti","shruti170799@gmail.com",f1);
                Contact Sakshi = new Contact("Sakshi","sakshi@gmail.com",f1);
                Contact Rohit = new Contact("Rohit","rohit@gmail.com",m1);
                Contact Amit = new Contact("Amit","amit@gmail.com",m1);

                contact.put((long) 802819281, Shruti);
                contact.put((long) 987291792 , Sakshi);
                contact.put((long) 896273627, Rohit);
                contact.put((long) 902617628, Amit);

                Set set = contact.entrySet();
                Iterator i = set.iterator();

                while(i.hasNext()) {
                    Map.Entry me = (Map.Entry)i.next();
                    System.out.println("  Phone No. " + me.getKey());
                    System.out.println(   me.getValue());
                    System.out.println("  Phone No. " + me.getKey() + me.getValue());
                }
```

a)Fetch keys:

```
Problems  Javadoc  Declaration  Consol
<terminated> Main (7) [Java Application] C:\Program
  Phone No. 987291792
  Phone No. 902617628
  Phone No. 896273627
  Phone No. 802819281
```

b)Fetch Values:

```
Problems  Javadoc  Declaration  Console  Error Log
<terminated> Main (7) [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe
Contact [name=Sakshi,email=sakshi@gmail.com,gender=Female]
Contact [name=Amit,email=amit@gmail.com,gender=Male]
Contact [name=Rohit,email=rohit@gmail.com,gender=Male]
Contact [name=Shruti,email=shruti170799@gmail.com,gender=Female]
```

c)Fetch Keys and Values pair:

```
Problems  Javadoc  Declaration  Console  Error Log
<terminated> Main (7) [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe  (09-Aug-2021, 8:17:32 pm – 8:17:35 pm)
{987291792=Contact [name=Sakshi,email=sakshi@gmail.com,gender=Female], 902617628=Contact [name=Amit,email=amit@gmail.com,gender=Male],
```

```
Error Log
s\Java\jdk-13.0.2\bin\javaw.exe  (09-Aug-2021, 8:17:32 pm – 8:17:35 pm)
 896273627=Contact [name=Rohit,email=rohit@gmail.com,gender=Male], 802819281=Contact [name=Shruti,email=shruti170799@gmail.com,gender=Female]}
```

2)Duplicate Items:

```
1  import java.util.ArrayList;
4  public class Duplicate {
5
6      public static void main(String[] args) {
7
8          List<String> foodlist = new ArrayList<String>();
9          foodlist.add("Pizza");
10         foodlist.add("Burger");
11         foodlist.add("Fries");
12         foodlist.add("Samosa");
13         foodlist.add("Vadapav");
14         foodlist.add("Kachori");
15         foodlist.add("Sweet");
16         foodlist.add("Dhokla");
17         foodlist.add("Gulabjamun");
18         foodlist.add("Khandvi");
19         //trying to add Duplicate Item
20         foodlist.add("Fries");
21         //duplicate item will not be printed in Treeset
22         TreeSet<String> treeset=new TreeSet<String>(foodlist);
23         System.out.println(treeset);
24
25     }
26
27 }
```

Output:

```
Problems  Javadoc  Declaration  Console  Error Log
<terminated> Duplicate [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe  (09-Aug-2021, 8:18:59 pm –
[Burger, Dhokla, Fries, Gulabjamun, Kachori, Khandvi, Pizza, Samosa, Sweet, Vadapav]
```

3)Employee Sorting:

a)Employee class:

Collection ▸ src ▸ (default package) ▸ Employye ▸ Employye(int, String, String, int)

```java
import java.util.*;

class Employye {
    int id;
    String name;
    String Department;
    int Salary;

    public Employye(int id, String name, String Department, int 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 int getSalary() {
        return Salary;
    }

    public void setSalary(int Salary) {
        this.Salary = Salary;
    }

    @Override
    public String toString() {
        return  "" + this.id + " " +this.name + " " +this.Department + " " +this.Salary;
    }
}
```

b)sort class:

```java
import java.io.*;
class Sort {

    public static void main(String[] args) {

            TreeSet<Employye> employees = new TreeSet<>(new FirstComparator());

            employees.add(new Employye(8,"Shruti","IT",210000));
            employees.add(new Employye(3,"Rohit","COMP",240000));
            employees.add(new Employye(5,"Sam","EXTC",300000));
            employees.add(new Employye(4,"Akash","MECH",100000));
            employees.add(new Employye(6,"Yogesh","IT",250000));
            employees.add(new Employye(10,"Aish","ART",320000));
            employees.add(new Employye(2,"Priti","IT",400000));
            employees.add(new Employye(1,"Fheba","SCIENCE",150000));
            employees.add(new Employye(7,"Binoy","IT",3500000));
            employees.add(new Employye(9,"Romax","EXTC",380000));

            Scanner sc=new Scanner(System.in);
            System.out.println("========================================");
            System.out.print("Enter Your Choice: ");
            String str = sc.next();
            String a1="b";
            Iterator<Employye> it = employees.iterator();
            if(a1.equals(a1)) {
            for(Employye emp : employees) {
            System.out.println(emp.getId() + " " +emp.getName() + " " +emp.getDepartment() + " "+emp.getSalary() );
                }
            System.out.println("Sorted on the basis of Id");


            TreeSet<Employye> employees1 = new TreeSet<>(new Name());


                employees1.add(new Employye(8,"Shruti","IT",1349202));
                employees1.add(new Employye(3,"Rohit","COMP",240000));
                employees1.add(new Employye(5,"Sam","EXTC",300000));
```

```
                employees1.add(new Employye(4,"Akash","MECH",100000));
                employees1.add(new Employye(6,"Yogesh","IT",250000));
                employees1.add(new Employye(10,"Aish","ART",320000));
                employees1.add(new Employye(2,"Priti","IT",400000));
                employees1.add(new Employye(1,"Fheba","SCIENCE",150000));
                employees1.add(new Employye(7,"Binoy","IT",3500000));
                employees1.add(new Employye(9,"Romax","EXTC",380000));

                Scanner sc1=new Scanner(System.in);
                 System.out.println("=========================================");
                System.out.print("Enter Your Choice: ");
                String str1 = sc.next();
                String a2="b";
                Iterator<Employye> it1 =employees1.iterator();
                if(a2.equals(a2)) {
                for(Employye emp : employees1) {
                System.out.println(emp.getId() + " " +emp.getName() + " " +emp.getDepartment() + " "+emp.getSalary() );
                }

                System.out.println("Sorted on the basis of name");

                  TreeSet<Employye> employees2 = new TreeSet<>(new department());


                    employees2.add(new Employye(8,"Shruti","IT",210000));
                    employees2.add(new Employye(3,"Rohit","COMP",240000));
                    employees2.add(new Employye(5,"Sam","EXTC",300000));
                    employees2.add(new Employye(4,"Akash","MECH",100000));
                    employees2.add(new Employye(6,"Yogesh","HR",250000));
                    employees2.add(new Employye(10,"Aish","BU",320000));
                    employees2.add(new Employye(2,"Priti","TESTING",400000));
                    employees2.add(new Employye(1,"Fheba","DEV",150000));
                    employees2.add(new Employye(7,"Binoy","AI",3500000));
                    employees2.add(new Employye(9,"Romax","ML",380000));


                Scanner sc2=new Scanner(System.in);
                System.out.println("=========================================");
                System.out.print("Enter Your Choice: ");
                String str2 = sc.next();
                String a3="c";
                Iterator<Employye> it2 =employees2.iterator();
                if(a3.equals(a3)) {
                for(Employye emp : employees2) {
                System.out.println(emp.getId() + " " +emp.getName() + " " +emp.getDepartment() + " "+emp.getSalary() );
                }
                System.out.println("Sorted on the basis of Department");


                TreeSet<Employye> employees3 = new TreeSet<>(new salary());


                employees3.add(new Employye(8,"Shruti","IT",210000));
                employees3.add(new Employye(3,"Rohit","COMP",240000));
                employees3.add(new Employye(5,"Sam","EXTC",300000));
                employees3.add(new Employye(4,"Akash","MECH",100000));
                employees3.add(new Employye(6,"Yogesh","IT",250000));
                employees3.add(new Employye(10,"Aish","ART",320000));
                employees3.add(new Employye(2,"Priti","IT",400000));
                employees3.add(new Employye(1,"Fheba","SCIENCE",150000));
                employees3.add(new Employye(7,"Binoy","IT",3500000));
                employees3.add(new Employye(9,"Romax","EXTC",380000));

                Scanner sc3=new Scanner(System.in);
                System.out.println("=========================================");
                System.out.print("Enter Your Choice: ");
                String str3 = sc.next();
                String a4="d";
                Iterator<Employye> it3 =employees3.iterator();
                if(a4.equals(a4)) {
                for(Employye emp : employees3) {
                System.out.println(emp.getId() + " " +emp.getName() + " " +emp.getDepartment() + " "+emp.getSalary() );

                 }
                 System.out.println("Sorted on the basis of Salary");
                 }
                 }
                }
            }
    }


}
```

c)Comparator classes:

```java
import java.util.Comparator;
class Name implements Comparator<Employye>{

    @Override
    public int compare(Employye e1, Employye e2) {

        return (e1.name).compareTo(e2.name);

}}
```

```java
import java.util.Comparator;
class FirstComparator implements Comparator<Employye>{

    @Override
    public int compare(Employye e1, Employye e2) {
        // TODO Auto-generated method stub
        {
            if(e2.id > e1.id) {
                return -1;
            }
            else if (e2.id < e1.id) {
                return 1;
            }
            else {
                return Integer.compare(e2.getId(), e1.getId());
            }
    }
}
}
```

```java
import java.util.Comparator;
import java.util.Iterator;
class department implements Comparator<Employye>{

    @Override
    public int compare(Employye e1, Employye e2) {

        return (e1.Department).compareTo(e2.Department);

}

}

```

```java
import java.util.Comparator;
class salary implements Comparator<Employye>{

    @Override
    public int compare(Employye e1, Employye e2) {
        // TODO Auto-generated method stub
        {
            if(e2.Salary > e1.Salary) {
                return -1;
            }
            else if (e2.Salary < e1.Salary) {
                return 1;
            }
            else {
                return Integer.compare(e2.getSalary(), e1.getSalary());
            }
    }
}
}

```

Output:

Problems Javadoc Declaration Console Error Log

<terminated> Sort [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe (09-Aug-2

```
==========================================
Enter Your Choice: a
1 Fheba SCIENCE 150000
2 Priti IT 400000
3 Rohit COMP 240000
4 Akash MECH 100000
5 Sam EXTC 300000
6 Yogesh IT 250000
7 Binoy IT 3500000
8 Shruti IT 210000
9 Romax EXTC 380000
10 Aish ART 320000
Sorted on the basis of Id
==========================================
Enter Your Choice: b
10 Aish ART 320000
4 Akash MECH 100000
7 Binoy IT 3500000
1 Fheba SCIENCE 150000
2 Priti IT 400000
3 Rohit COMP 240000
9 Romax EXTC 380000
5 Sam EXTC 300000
8 Shruti IT 1349202
6 Yogesh IT 250000
Sorted on the basis of name
------------------------------------------

Enter Your Choice: c
7 Binoy AI 3500000
10 Aish BU 320000
3 Rohit COMP 240000
1 Fheba DEV 150000
5 Sam EXTC 300000
6 Yogesh HR 250000
8 Shruti IT 210000
4 Akash MECH 100000
9 Romax ML 380000
2 Priti TESTING 400000
Sorted on the basis of Department
==========================================
Enter Your Choice: d
4 Akash MECH 100000
1 Fheba SCIENCE 150000
8 Shruti IT 210000
3 Rohit COMP 240000
6 Yogesh IT 250000
5 Sam EXTC 300000
10 Aish ART 320000
9 Romax EXTC 380000
2 Priti IT 400000
7 Binoy IT 3500000
Sorted on the basis of Salary
```

4)Date Of Birth (leap Year):

```
1  import java.util.ArrayList;
8
9  public class Date {
10
11     public static void main(String[] args) {
12         {
13
14         LocalDate date1 = LocalDate.of(2000, 12, 23);
15         LocalDate date2 = LocalDate.of(2001, 12, 23);
16
17         Collection<Object> obj=new LinkedList<>();
18         obj.add(date1);
19         obj.add(date2);
20         for(Object i: obj) {
21
22             int a,c;
23             int y=date1.getYear();
24             int y1=date2.getYear();
25             if(y!=0)
26             {
27             a=(y%400==0)?(c=1):(y%100==0)?(c=0):((y%4==0)?(c=1):(c=0));
28             if(a==1)
29                 System.out.println("Your Date of Birth is " +date1+" and it was a leap year");
30             else
31                 System.out.println("Your Date of Birth is " +date1+" and it was not a leap year");
32             if(y1!=0)
33             {
34             a=(y1%400==0)?(c=1):(y1%100==0)?(c=0):((y1%4==0)?(c=1):(c=0));
35             if(a==1)
36                 System.out.println("Your Date of Birth is " +date2+" and it was a leap year");
37             else
38                 System.out.println("Your Date of Birth is " +date2+" and it was not a leap year");
39
40
41             Iterator<Object> itr=obj.iterator();
42             while(itr.hasNext()) {
43                             }
44
45             }
46         }
47         }
48     }
49 }
50 }
51
```

Output:

```
Problems  Javadoc  Declaration  Console  Error Log
Date [Java Application] C:\Program Files\Java\jdk-13.0.2\bin\javaw.exe  (09-Aug-2021,
Your Date of Birth is 2000-12-23 and it was a leap year
Your Date of Birth is 2001-12-23 and it was not a leap year
```