2.

```java
1
2
3⊕    import java.util.*;☐
5
6     public class HashSet1 {
7
8⊖        public static void main(String[] args)
9         {
10            // Instantiate an object
11            // of HashSet
12            HashSet<String> hs = new HashSet<String>();
13
14            // Elements are added using add() method
15            hs.add("Riddhi");
16            hs.add("pooja");
17            hs.add("tanaya");
18            hs.add("sakshi");
19            hs.add("shruti");
20            hs.add("raj");
21            hs.add("vivek");
22            hs.add("shubham");
23            hs.add("bhoomi");
24            hs.add("Riddhi");
25
26
27
28
29            System.out.println("HashSet elements : " + hs + " " + " : Duplicate values are ignored");
30        }
31    }
32
33
```
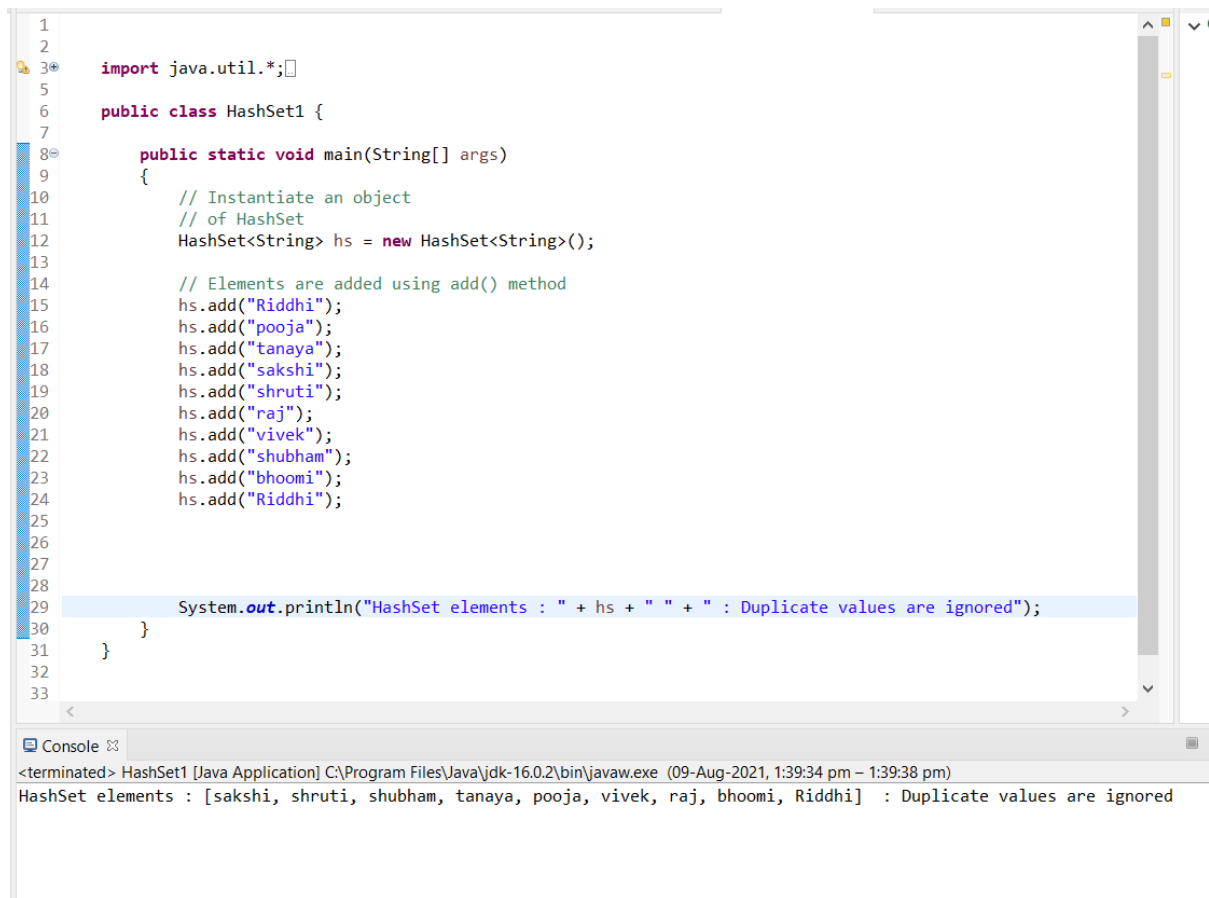
Console ✕

<terminated> HashSet1 [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (09-Aug-2021, 1:39:34 pm – 1:39:38 pm)

```
HashSet elements : [sakshi, shruti, shubham, tanaya, pooja, vivek, raj, bhoomi, Riddhi]  : Duplicate values are ignored
```

1.

```java
import java.util.*;
class Contact {

String name;
String email;
String gender;

public Contact(String name, String email, String gender) {

    this.name = name;
    this.email = email;
    this.gender = gender;

}



}

public class MapExample {
public static void main(String[] args) {
    //Creating map of Contacts
```

```java
    Map<Long,Contact>contact =new
TreeMap<Long,Contact>(Collections.reverseOrder());
    //Creating Contact

    //Adding Contact to map
    String name;
        String email;
        String gender;
        Contact Riddhi = new Contact(name = "Riddhi", email= "riddhi@gmail.com",
gender= "Female");
        contact.put((long) 854172367, Riddhi);


    //Traversing map
    for(Map.Entry<Long, Contact> entry:contact.entrySet()){
        long key=entry.getKey();
        Contact c=entry.getValue();
        System.out.println(key+" Details:");
        System.out.println(c.name+" "+c.email+" "+c.gender+" ");
    }
}
}
```
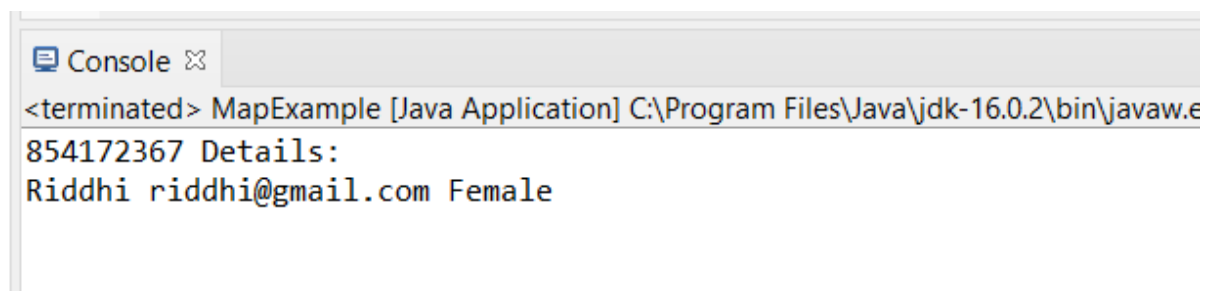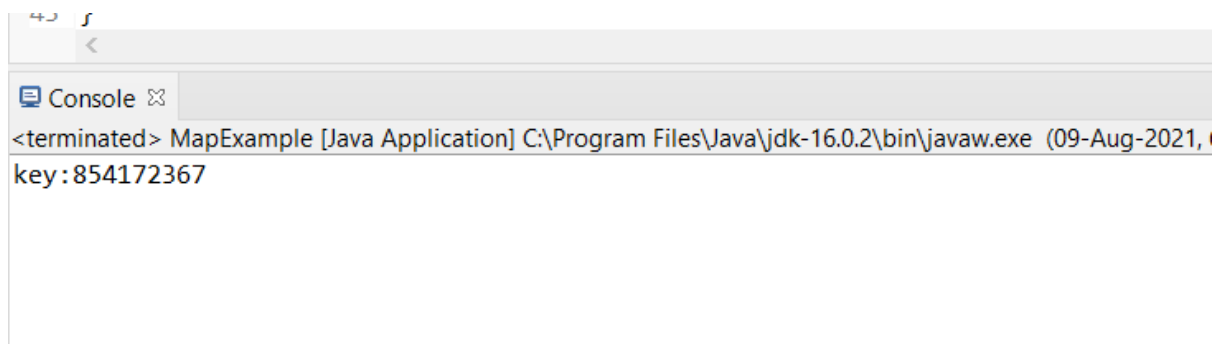
Console ⊠

<terminated> MapExample [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.e

```
854172367 Details:
Riddhi riddhi@gmail.com Female
```

Console ⊠

<terminated> MapExample [Java Application] C:\Program Files\Java\jdk-16.0.2\bin\javaw.exe (09-Aug-2021,

```
key:854172367
```

3.

```java
import java.util.*;

class Employee implements Comparable<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() {
```

```java
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    @Override
    public int hashCode() {
    return Objects.hash(id, name, department, salary);
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj)
            return true;
        if (obj == null || getClass() != obj.getClass())
            return false;
        Employee employee = (Employee) obj;
        return Objects.equals(department, employee.department) && id==employee.id
                    && Objects.equals(name, employee.name) &&
salary==employee.salary;
    }

        @Override
        public int compareTo(Employee e) {
            return this.getId() - e.getId();
        }

        @Override
        public String toString() {
            return ("Employee [" +"Id=" + id +", " +"Name=" + name + ", "
+"Department="+department+ ", "
                        +"Salary= "+salary+"]");
        }
 }


import java.util.Comparator;
import java.util.Iterator;
import java.util.Scanner;
import java.util.Set;
import java.util.TreeSet;

public class Collection3 {
public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    String choice;
    System.out.println("Run Application: \n a)Id \n b)Name \n c)Department \n
d)Salary");
    System.out.print("Enter Your Choice: ");
    choice= in.next();
    in.close();

    Set<Employee> values = new TreeSet<>();

    values.add(new Employee(101, "Riddhi", "IT",300000));
    values.add(new Employee(102, "Siddhi", "ET",400000));
    values.add(new Employee(103, "Niddhi", "HR", 540000));
```

```java
        values.add(new Employee(104, "Shubham", "IT",200000));
        values.add(new Employee(105, "Raj", "Training",750000));
        values.add(new Employee(106, "Ramesh", "CSE", 123000));
        values.add(new Employee(107, "soniya", "ELECTRICAL",350000));
        values.add(new Employee(108, "karan", "MECH",980000));
        values.add(new Employee(109, "kartik","MS", 850000));
        values.add(new Employee(110, "Bhoomi","MT", 550000));

        if (choice.equals("a")) {
           Iterator<Employee> it= values.iterator();
           while(it.hasNext()) {
                 System.out.println(it.next());
           }
        }
        else if (choice.equals("b")) {
           values = new TreeSet<>(Comparator.comparing(Employee::getName));

           values.add(new Employee(101, "Riddhi", "IT",300000));
            values.add(new Employee(102, "Siddhi", "ET",400000));
            values.add(new Employee(103, "Niddhi", "HR", 540000));
            values.add(new Employee(104, "Shubham", "IT",200000));
            values.add(new Employee(105, "Raj", "Training",750000));
            values.add(new Employee(106, "Ramesh", "CSE", 123000));
            values.add(new Employee(107, "soniya", "ELECTRICAL",350000));
            values.add(new Employee(108, "karan", "MECH",980000));
            values.add(new Employee(109, "kartik","MS", 850000));
            values.add(new Employee(110, "Bhoomi","MT", 550000));
            Iterator<Employee> it= values.iterator();
           while(it.hasNext()) {
                 System.out.println(it.next());
           }
        }

        else if(choice.equals("c")) {
           values = new TreeSet<>(Comparator.comparing(Employee::getDepartment));

           values.add(new Employee(101, "Riddhi", "IT",300000));
            values.add(new Employee(102, "Siddhi", "ET",400000));
            values.add(new Employee(103, "Niddhi", "HR", 540000));
            values.add(new Employee(104, "Shubham", "IT",200000));
            values.add(new Employee(105, "Raj", "Training",750000));
            values.add(new Employee(106, "Ramesh", "CSE", 123000));
            values.add(new Employee(107, "soniya", "ELECTRICAL",350000));
            values.add(new Employee(108, "karan", "MECH",980000));
            values.add(new Employee(109, "kartik","MS", 850000));
            values.add(new Employee(110, "Bhoomi","MT", 550000));
            Iterator<Employee> it= values.iterator();
           while(it.hasNext()) {
                 System.out.println(it.next());
           }
        }

        else if(choice.equals("d")) {
           values = new TreeSet<>(Comparator.comparing(Employee::getSalary));

           values.add(new Employee(101, "Riddhi", "IT",300000));
            values.add(new Employee(102, "Siddhi", "ET",400000));
            values.add(new Employee(103, "Niddhi", "HR", 540000));
            values.add(new Employee(104, "Shubham", "IT",200000));
```
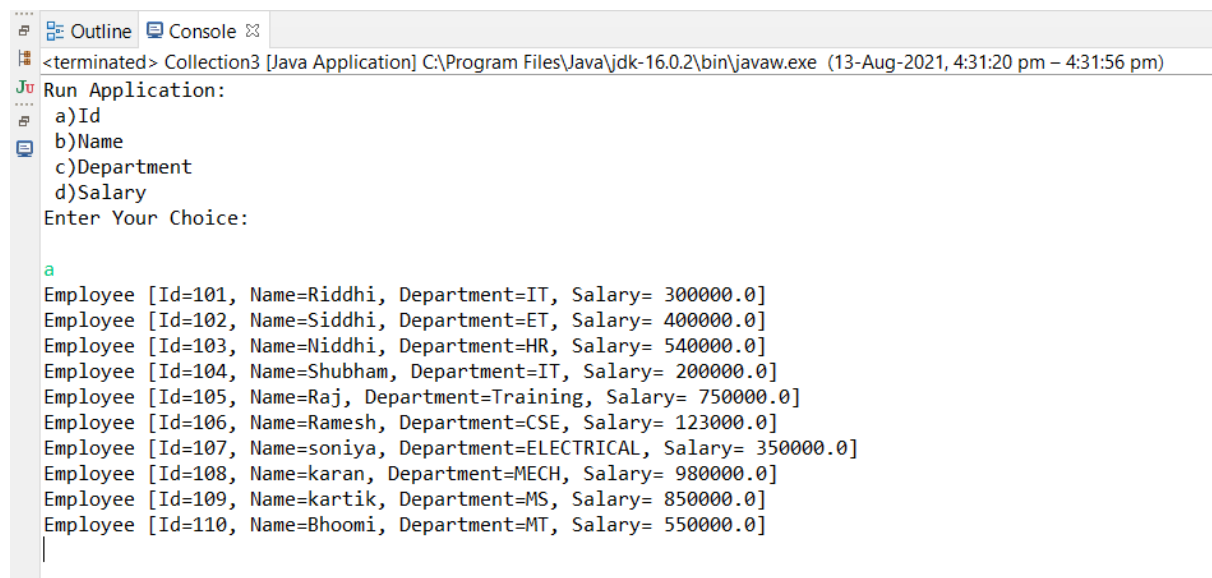
```java
        values.add(new Employee(105, "Raj", "Training",750000));
        values.add(new Employee(106, "Ramesh", "CSE", 123000));
        values.add(new Employee(107, "soniya", "ELECTRICAL",350000));
        values.add(new Employee(108, "karan", "MECH",980000));
        values.add(new Employee(109, "kartik","MS", 850000));
        values.add(new Employee(110, "Bhoomi","MT", 550000));
        Iterator<Employee> it= values.iterator();
    while(it.hasNext()) {
        System.out.println(it.next());
    }
  }
 }
}
```

Output:

```
Run Application:
 a)Id
 b)Name
 c)Department
 d)Salary
Enter Your Choice:

a
Employee [Id=101, Name=Riddhi, Department=IT, Salary= 300000.0]
Employee [Id=102, Name=Siddhi, Department=ET, Salary= 400000.0]
Employee [Id=103, Name=Niddhi, Department=HR, Salary= 540000.0]
Employee [Id=104, Name=Shubham, Department=IT, Salary= 200000.0]
Employee [Id=105, Name=Raj, Department=Training, Salary= 750000.0]
Employee [Id=106, Name=Ramesh, Department=CSE, Salary= 123000.0]
Employee [Id=107, Name=soniya, Department=ELECTRICAL, Salary= 350000.0]
Employee [Id=108, Name=karan, Department=MECH, Salary= 980000.0]
Employee [Id=109, Name=kartik, Department=MS, Salary= 850000.0]
Employee [Id=110, Name=Bhoomi, Department=MT, Salary= 550000.0]
```

```
Run Application:
 a)Id
 b)Name
 c)Department
 d)Salary
Enter Your Choice: b
Employee [Id=110, Name=Bhoomi, Department=MT, Salary= 550000.0]
Employee [Id=108, Name=Karan, Department=MECH, Salary= 980000.0]
Employee [Id=109, Name=Kartik, Department=MS, Salary= 850000.0]
Employee [Id=103, Name=Niddhi, Department=HR, Salary= 540000.0]
Employee [Id=105, Name=Raj, Department=Training, Salary= 750000.0]
Employee [Id=106, Name=Ramesh, Department=CSE, Salary= 123000.0]
Employee [Id=101, Name=Riddhi, Department=IT, Salary= 300000.0]
Employee [Id=104, Name=Shubham, Department=IT, Salary= 200000.0]
Employee [Id=102, Name=Siddhi, Department=ET, Salary= 400000.0]
Employee [Id=107, Name=Soniya, Department=ELECTRICAL, Salary= 350000.0]
```

```
Run Application:
 a)Id
 b)Name
 c)Department
 d)Salary
Enter Your Choice: d
Employee [Id=106, Name=Ramesh, Department=CSE, Salary= 123000.0]
Employee [Id=104, Name=Shubham, Department=IT, Salary= 200000.0]
Employee [Id=101, Name=Riddhi, Department=IT, Salary= 300000.0]
Employee [Id=107, Name=soniya, Department=ELECTRICAL, Salary= 350000.0]
Employee [Id=102, Name=Siddhi, Department=ET, Salary= 400000.0]
Employee [Id=103, Name=Niddhi, Department=HR, Salary= 540000.0]
Employee [Id=110, Name=Bhoomi, Department=MT, Salary= 550000.0]
Employee [Id=105, Name=Raj, Department=Training, Salary= 750000.0]
Employee [Id=109, Name=kartik, Department=MS, Salary= 850000.0]
Employee [Id=108, Name=karan, Department=MECH, Salary= 980000.0]
```

4.

```java
import java.time.LocalDate;
import java.util.*;
public class Collection4 {

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

```java
        LocalDate date1 = LocalDate.of(1999, 04, 04);
        LocalDate date2 = LocalDate.of(2000, 12, 27);

        Collection<Object> obj=new LinkedList<>();
    obj.add(date1);
    obj.add(date2);
    for(Object i: obj) {
        int a,c;
        int y1=date1.getYear();
        int y2=date2.getYear();

        if(y1!=0) {
         a=(y1%400==0)?(c=1):((y1%100==0)?(c=0):((y1%4==0)?(c=1):(c=0)));
         if(a==1) {
            System.out.println("Your date of birth is "+date1+ " and it
was a leap year.");
            }
            else {
                System.out.println("Your date of birth is "+date1+ "
and it was not a leap year.");
                }

            if(y2!=0) {
            a=(y2%400==0)?(c=1):((y2%100==0)?(c=0):((y2%4==0)?(c=1):(c=0)));
            if(a==1) {
            System.out.println("Your date of birth is "+date2+ " and it was
a leap year.");
            }
            else {
                System.out.println("Your date of birth is "+date2+ " and it
was not a leap year.");

            }

            Iterator<Object> it=obj.iterator();
            while(it.hasNext()) {
                //System.out.println(it.next());
                }
        }
        }
    }
    }
}
```
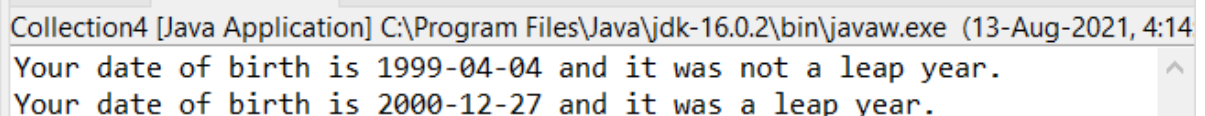
Output:

```
Your date of birth is 1999-04-04 and it was not a leap year.
Your date of birth is 2000-12-27 and it was a leap year.
```