```
----------TestVector.java---------
package com.ameya.test;

import java.util.Vector;

public class TestVector {

    public static void main(String[] args) {
            Vector<Integer> v=new Vector<Integer>(5,2);
            // size-->current no. elements in vector
            //capacity--> capacity of the vector
            System.out.println("SIZE :: "+v.size());
            System.out.println("CAPACITY :: "+v.capacity());
            v.addElement(10);
            v.addElement(20);
            v.addElement(30);
            v.addElement(40);
            v.addElement(50);
            System.out.println("SIZE :: "+v.size());
            System.out.println("CAPACITY :: "+v.capacity());
        v.addElement(10);
        v.addElement(20);
            v.addElement(30);
        //  v.addElement(40);
        //  v.addElement(50);
            System.out.println("SIZE :: "+v.size());
        System.out.println("CAPACITY :: "+v.capacity());
        //  v.addElement(50);
        //  System.out.println("SIZE :: "+v.size());
        //  System.out.println("CAPACITY :: "+v.capacity());
    }
}
-------------TestHashSet.java---------
package com.ameya.test;

import java.util.HashSet;
import java.util.Iterator;

import com.ameya.domain.Person;

public class TestHashSet {

    public static void main(String[] args) {
        HashSet<Person> hs=new HashSet<Person>();
        hs.add(new Person(5, "aaaa", "aaaa", 25));
        hs.add(new Person(3, "bbbb", "bbbb", 26));
        hs.add(new Person(4, "cccc", "cccc", 24));
        hs.add(new Person(1, "dddd", "dddd", 28));
        hs.add(new Person(2, "eeee", "eeee", 27));
        hs.add(new Person(1, "ffff", "ffff", 28));//Silently ignored
        Iterator<Person> itr=hs.iterator();
        while(itr.hasNext()) {
            Person person=itr.next();
            System.out.print(person);
        }
    }

}
-------------TestTreeSet.java---------
package com.ameya.test;

import java.util.Iterator;
import java.util.TreeSet;

public class TestTreeSet {

    public static void main(String[] args) {
        TreeSet<Integer> ts=new TreeSet<Integer>();
        ts.add(5);
        ts.add(3);
        ts.add(4);
        ts.add(1);
        ts.add(2);
        System.out.println(ts);
        System.out.println("====================================");
        Iterator<Integer> itr=ts.iterator();
        while(itr.hasNext()) {
            int n=itr.next();
            System.out.println(n);
        }
        System.out.println("====================================");
        TreeSet<Integer> revTs=(TreeSet<Integer>) ts.descendingSet();
        System.out.println(revTs);
        System.out.println("===============================");
        Iterator<Integer> itr1=revTs.iterator();
        while(itr1.hasNext()) {
            int n=itr1.next();
            System.out.println(n);
        }
    }

}
--------------Person.java---------
package com.ameya.domain;

public class Person implements Comparable<Person>{
    private long id;
    private String firstName;
    private String lastName;
    private int age;
    public Person() {
        super();
        // TODO Auto-generated constructor stub
    }
    public Person(long id, String firstName, String lastName, int age) {
        super();
        this.id = id;
        this.firstName = firstName;
        this.lastName = lastName;
        this.age = age;
    }
    public long getId() {
        return id;
    }
    public void setId(long id) {
        this.id = id;
    }
    public String getFirstName() {
        return firstName;
    }
    public void setFirstName(String firstName) {
        this.firstName = firstName;
    }
```

```java
    public String getLastName() {
        return lastName;
    }
    public void setLastName(String lastName) {
        this.lastName = lastName;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    @Override
    public String toString() {
        return "Person [id=" + id + ", firstName=" + firstName + ", lastName=" +
    lastName + ", age=" + age + "]\n";
    }
    @Override
    public boolean equals(Object obj) {
        return this.id==((Person)obj).getId() ? true : false;
    }
    @Override
    public int hashCode() {
        final long prime=31;
        long result=1;
        result=prime*result+id;
        return (int)result;
    }
    @Override
    public int compareTo(Person o) {
        return ((int)(this.id-o.getId()));
    }
}
--------------TestPersonTreeSet.java---------
package com.ameya.test;

import java.util.Iterator;
import java.util.TreeSet;

import com.ameya.domain.Person;
```

```java

public class TestPersonTreeSet {

    public static void main(String[] args) {
        TreeSet<Person> ts=new TreeSet<Person>();
        ts.add(new Person(5, "aaaa", "aaaa", 25));
        ts.add(new Person(3, "bbbb", "bbbb", 26));
        ts.add(new Person(4, "cccc", "cccc", 24));
        ts.add(new Person(1, "dddd", "dddd", 28));
        ts.add(new Person(2, "eeee", "eeee", 27));
        System.out.println(ts);
        System.out.println("====================================");
        Iterator<Person> itr=ts.iterator();
        while(itr.hasNext()) {
            Person person=itr.next();
            System.out.print(person);
        }
        System.out.println("\n=====================================");
        TreeSet<Person> rev=(TreeSet<Person>) ts.descendingSet();
        Iterator<Person> itr1=rev.iterator();
                System.out.println(rev);
        System.out.println("=========================================");
        while(itr1.hasNext()) {
            Person person=itr1.next();
            System.out.print(person);
        }
    }

}
```