```
1
2   ======================Person.java============================
3   package com.ameya.models;
4
5   package com.ameya.models;
6
7   public class Person {
8       private int id;
9       private String firstName;
10      private String lastName;
11      private int age;
12      public Person() {
13          firstName="N/A";
14          lastName="N/A";
15          id=-1;
16          age=-1;
17      }
18      public Person(int id, String firstName, String lastName, int age) {
19          super();
20          this.id = id;
21          this.firstName = firstName;
22          this.lastName = lastName;
23          this.age = age;
24      }
25      public int getId() {
26          return id;
27      }
28      public void setId(int id) {
29          this.id = id;
30      }
31      public String getFirstName() {
32          return firstName;
33      }
34      public void setFirstName(String firstName) {
35          this.firstName = firstName;
36      }
37      public String getLastName() {
38          return lastName;
39      }
40      public void setLastName(String lastName) {
41          this.lastName = lastName;
42      }
43      public int getAge() {
44          return age;
45      }
46      public void setAge(int age) {
47          this.age = age;
48      }
49      @Override
50      public String toString() {
51          return "Person [id=" + id + ", firstName=" + firstName + ", lastName=" +
            lastName + ", age=" + age + "]\n";
52      }
53
54      @Override
55      public int hashCode() {
56          final int prime = 31;
57          int result = 1;
58          result = prime * result + id;
59          return result;
60      }
61      @Override
62      public boolean equals(Object obj) {
63          Person p=(Person)obj;
64          return this.id==p.getId()?true:false;
65      }
66
67
68
69  }
70  ============================HashSetUtil.java==============================
71  package com.ameya.util;
72
```

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

import com.ameya.models.Person;

public class HashSetUtil {
    private HashSet<Person> hs;
    public HashSetUtil() {
        hs=new HashSet<>();
    }
    public void addPerson(Person person) {
        hs.add(person);
    }
    public void showAllPersons() {
        Iterator<Person> itr=hs.iterator();
        while(itr.hasNext()) {
            Person person=itr.next();
            System.out.println(person);
        }
    }
}


============================TestHashSetUtil.java============================
package com.ameya.test;

import com.ameya.models.Person;
import com.ameya.util.HashSetUtil;

public class TestHashSetUtil {

    public static void main(String[] args) {
        HashSetUtil hsObj=new HashSetUtil();
        hsObj.addPerson(new Person(5,"aaaa","aaaa",25));
        hsObj.addPerson(new Person(3,"bbbb","bbbb",26));
        hsObj.addPerson(new Person(4,"cccc","cccc",24));
        hsObj.addPerson(new Person(1,"dddd","dddd",27));
        hsObj.addPerson(new Person(2,"eeee","eeee",28));
        hsObj.addPerson(new Person(1,"ffff","ffff",27));
        hsObj.showAllPersons();

    }

}
==================================HashMapUtil.java==================================
package com.ameya.util;

import java.util.HashMap;
import java.util.Iterator;
import java.util.Map;

import com.ameya.models.Person;

public class HashMapUtil {
    private HashMap<Integer, Person> map;

    public HashMapUtil() {
        map=new HashMap<>();
    }
    public void addPerson(Person person) {
        map.put(person.getId(), person);
    }
    public void traverseByKeys() {
        Iterator<Integer> itr=map.keySet().iterator();
        while(itr.hasNext()) {
            int key=itr.next();
            Person person=map.get(key);
            System.out.println("KEY => "+key+" || VALUE =>"+person);
        }
    }
    public void traverseByEntries() {
        Iterator<Map.Entry<Integer, Person>> itr=map.entrySet().iterator();
```

```
145             while(itr.hasNext()) {
146                 Map.Entry<Integer, Person> entry=itr.next();
147                 System.out.println("KEY :: "+entry.getKey()+" || VALUE ::
                    "+entry.getValue());
148             }
149         }
150     }
151
152     ==================================TestHashMapUtil.java==================================
        ======
153     package com.ameya.test;
154
155     import com.ameya.models.Person;
156     import com.ameya.util.HashMapUtil;
157
158     public class TestHashMapUtil {
159
160         public static void main(String[] args) {
161             HashMapUtil hmObj=new HashMapUtil();
162             hmObj.addPerson(new Person(5,"aaaa","aaaa",25));
163             hmObj.addPerson(new Person(3,"bbbb","bbbb",26));
164             hmObj.addPerson(new Person(4,"cccc","cccc",24));
165             hmObj.addPerson(new Person(1,"dddd","dddd",27));
166             hmObj.addPerson(new Person(2,"eeee","eeee",28));
167             //hmObj.addPerson(new Person(1,"ffff","ffff",27));
168             hmObj.traverseByKeys();
169             System.out.println("<<<<<<<<<<<<<<<>>>>>>>>>>>>>>>>>>>");
170             hmObj.traverseByEntries();
171
172         }
173
174     }
175     ================================Person.java=========================================
        ===
176     package com.ameya.models;
177
178     public class Person implements Comparable<Person>{
179         private int id;
180         private String firstName;
181         private String lastName;
182         private int age;
183         public Person() {
184             firstName="N/A";
185             lastName="N/A";
186             id=-1;
187             age=-1;
188         }
189         public Person(int id, String firstName, String lastName, int age) {
190             super();
191             this.id = id;
192             this.firstName = firstName;
193             this.lastName = lastName;
194             this.age = age;
195         }
196         public int getId() {
197             return id;
198         }
199         public void setId(int id) {
200             this.id = id;
201         }
202         public String getFirstName() {
203             return firstName;
204         }
205         public void setFirstName(String firstName) {
206             this.firstName = firstName;
207         }
208         public String getLastName() {
209             return lastName;
210         }
211         public void setLastName(String lastName) {
212             this.lastName = lastName;
213         }
214         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 int hashCode() {
        final int prime = 31;
        int result = 1;
        result = prime * result + id;
        return result;
    }

    @Override
    public boolean equals(Object obj) {
        Person p=(Person)obj;
        return this.id==p.getId()?true:false;
    }
    @Override
    public int compareTo(Person o) {
        /*
        if(this.id<o.getId()) {
            return -1;
        }else if(this.id>o.getId()) {
            return 1;
        }else {
            return 0;
        }*/
        return this.id-o.getId();
    }



}
==============================TreeSetUtil.java=======================================
======
package com.ameya.util;

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

import com.ameya.models.Person;

public class TreeSetUtil {

    private TreeSet<Integer> tsInt;
    private TreeSet<Person> tsPerson;

    public TreeSetUtil() {
        tsInt=new TreeSet<>();
        tsPerson=new TreeSet<>();
    }
    public void addIntegers(int num) {
        tsInt.add(num);
    }
    public void showIntegerTreeSet() {
        Iterator<Integer> itr=tsInt.iterator();
        while(itr.hasNext()) {
            int num=itr.next();
            System.out.println(num);
        }
    }
    public void addPerson(Person person) {
        tsPerson.add(person);
    }
    public void showPersonTreeSet() {
        Iterator<Person> itr=tsPerson.iterator();
```

```
286             while(itr.hasNext()) {
287                 Person person=itr.next();
288                 System.out.println(person);
289             }
290         }
291     }
292     ========================================TestTreeSetUtil.java========================
    ==================
293     package com.ameya.test;
294
295     import com.ameya.models.Person;
296     import com.ameya.util.TreeSetUtil;
297
298     public class TestTreeSetUtil {
299
300         public static void main(String[] args) {
301             TreeSetUtil tsObj=new TreeSetUtil();
302             tsObj.addIntegers(5);
303             tsObj.addIntegers(3);
304             tsObj.addIntegers(4);
305             tsObj.addIntegers(1);
306             tsObj.addIntegers(2);
307             tsObj.showIntegerTreeSet();
308             System.out.println("==============================================");
309             tsObj.addPerson(new Person(5,"aaaa","aaaa",25));
310             tsObj.addPerson(new Person(3,"bbbb","bbbb",26));
311             tsObj.addPerson(new Person(4,"cccc","cccc",24));
312             tsObj.addPerson(new Person(1,"dddd","dddd",27));
313             tsObj.addPerson(new Person(2,"eeee","eeee",28));
314             tsObj.showPersonTreeSet();
315
316         }
317
318     }
319     ======================================FirstNameComparator.java====================
    ==================
320     package com.ameya.util.comparators;
321
322     import java.util.Comparator;
323
324     import com.ameya.models.Person;
325
326     public class FirstNameComparator implements Comparator<Person>{
327
328         @Override
329         public int compare(Person o1, Person o2) {
330             return o1.getFirstName().compareTo(o2.getFirstName());
331         }
332
333     }
334     =======================================AgeComparator.java============================
    ============
335     package com.ameya.util.comparators;
336
337     import java.util.Comparator;
338
339     import com.ameya.models.Person;
340
341     public class AgeComparator implements Comparator<Person>{
342
343         @Override
344         public int compare(Person o1, Person o2) {
345             return o1.getAge()-o2.getAge();
346         }
347
348     }
349     ====================================TestComparators.java============================
    ===============
350     package com.ameya.test;
351
352     import java.util.ArrayList;
353     import java.util.Collections;
354
```

```
import com.ameya.models.Person;
import com.ameya.util.comparators.AgeComparator;
import com.ameya.util.comparators.FirstNameComparator;

public class TestComparators {

    public static void main(String[] args) {
        ArrayList<Person> list=new ArrayList<Person>();
        list.add(new Person(5,"cccc","cccc",22));
        list.add(new Person(3,"eeee","eeee",27));
        list.add(new Person(4,"aaaa","aaaa",23));
        list.add(new Person(1,"bbbb","bbbb",25));
        list.add(new Person(2,"dddd","dddd",24));
        System.out.println("-----------------------------------");
        System.out.println("List - No Sorting Criteria");
        System.out.println("-----------------------------------");
        System.out.println(list);
        System.out.println("-----------------------------------");
        System.out.println("List - Default Sorting Criteria - On ID");
        System.out.println("-----------------------------------");
        Collections.sort(list);
        System.out.println(list);
        System.out.println("-----------------------------------");
        System.out.println("List - Default Sorting Criteria - On ID - Descending");
        System.out.println("-----------------------------------");
        Collections.sort(list,Collections.reverseOrder());
        System.out.println(list);
        System.out.println("-----------------------------------");
        System.out.println("List - Sorted on FIRSTNAME - Ascending");
        System.out.println("-----------------------------------");
        Collections.sort(list,new FirstNameComparator());
        System.out.println(list);
        System.out.println("-----------------------------------");
        System.out.println("List - Sorted on FIRSTNAME - Descending");
        System.out.println("-----------------------------------");
        Collections.sort(list,Collections.reverseOrder(new FirstNameComparator()));
        System.out.println(list);
        System.out.println("-----------------------------------");
        System.out.println("List - Sorted on AGE - Ascending");
        System.out.println("-----------------------------------");
        Collections.sort(list,new AgeComparator());
        System.out.println(list);
        System.out.println("-----------------------------------");
        System.out.println("List - Sorted on AGE - Descending");
        System.out.println("-----------------------------------");
        Collections.sort(list,Collections.reverseOrder(new AgeComparator()));
        System.out.println(list);
    }

}
```