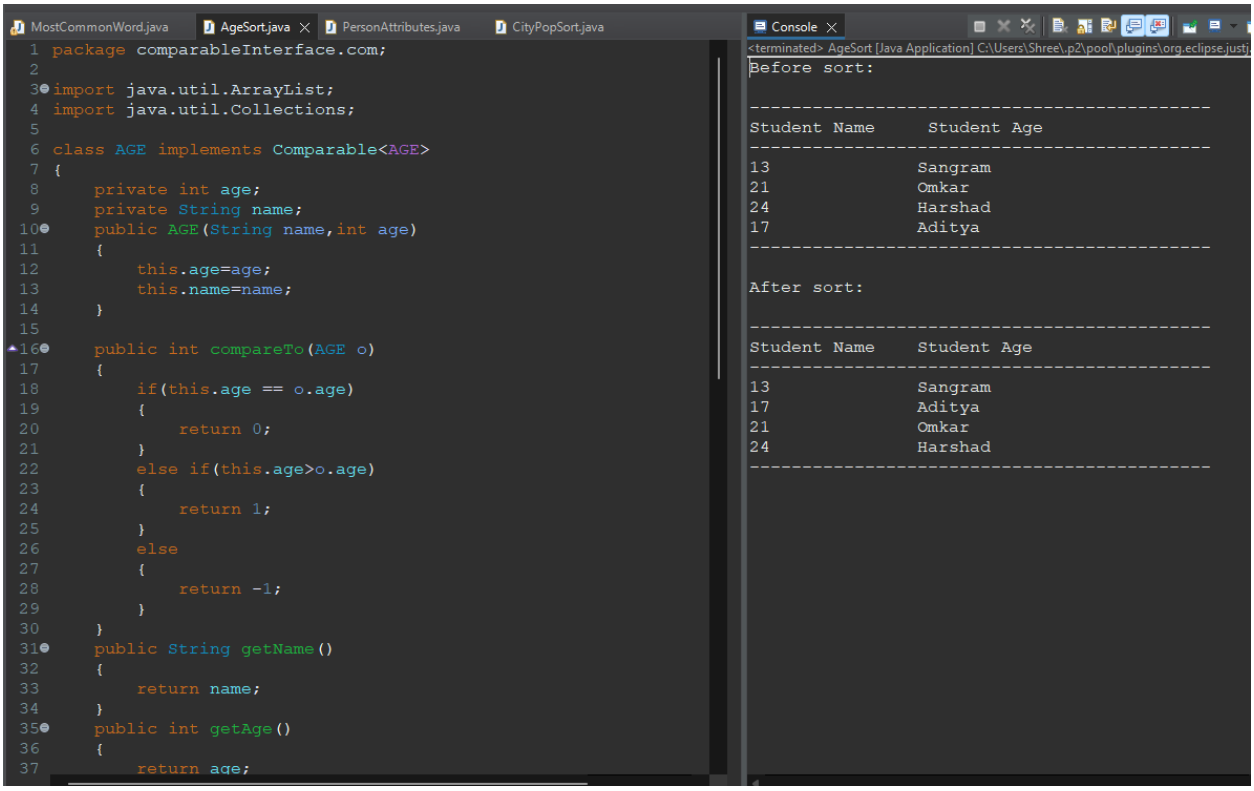
Assignment No:-55

Name:-Suryawanshi Sangramsingh Sambhaji

Batch: - Delta - DCA (Java) 2024 Date:-30/7/2024

Q.1 Implement the Comparable interface for a Person class that compares instances based on their age.



```java
package comparableInterface.com;

import java.util.ArrayList;
import java.util.Collections;

class AGE implements Comparable<AGE>
{
    private int age;
    private String name;
    public AGE(String name,int age)
    {
        this.age=age;
        this.name=name;
    }

    public int compareTo(AGE o)
    {
        if(this.age == o.age)
        {
            return 0;
        }
        else if(this.age>o.age)
        {
            return 1;
        }
        else
        {
            return -1;
        }
    }
    public String getName()
    {
        return name;
    }
    public int getAge()
    {
        return age;
```

```
Before sort:

-------------------------------------------
Student Name     Student Age
-------------------------------------------
13               Sangram
21               Omkar
24               Harshad
17               Aditya
-------------------------------------------

After sort:

-------------------------------------------
Student Name     Student Age
-------------------------------------------
13               Sangram
17               Aditya
21               Omkar
24               Harshad
-------------------------------------------
```

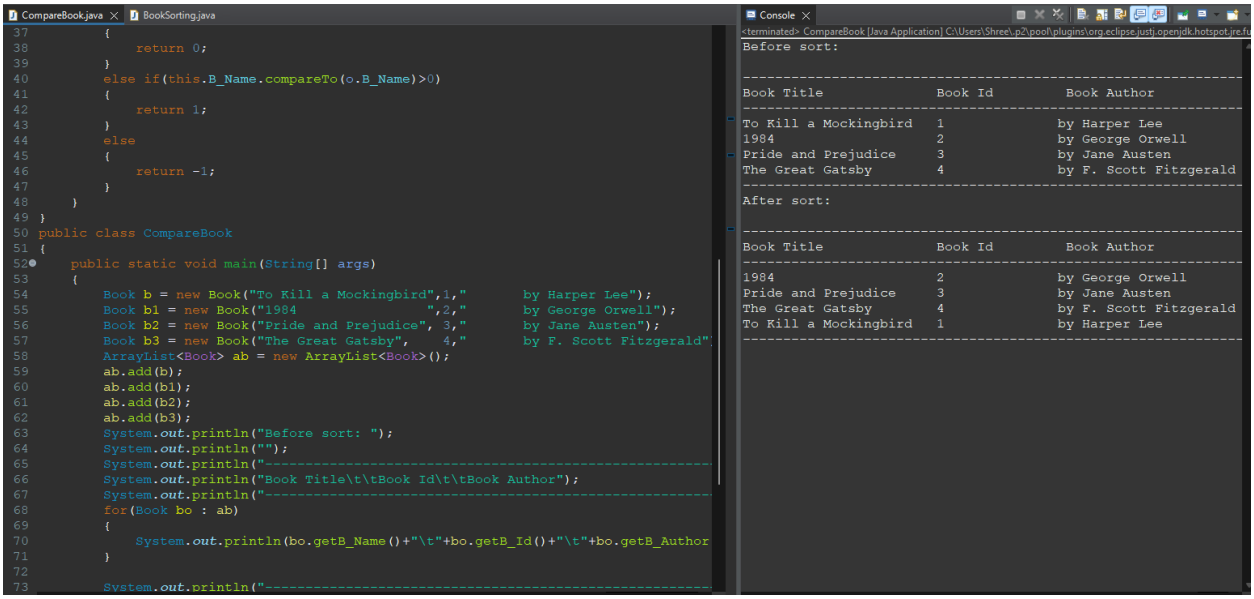
```
MostCommonWord.java | AgeSort.java | PersonAttributes.java | CityPopSort.java
38      }
39  }
40  public class AgeSort
41  {
42
43      public static void main(String[] args)
44      {
45          AGE a1 = new AGE("Sangram", 13);
46          AGE a2 = new AGE("Omkar", 21);
47          AGE a3 = new AGE("Harshad", 24);
48          AGE a4 = new AGE("Aditya", 17);
49          ArrayList<AGE> arr = new ArrayList<AGE>();
50          arr.add(a1);
51          arr.add(a2);
52          arr.add(a3);
53          arr.add(a4);
54          System.out.println("Before sort: ");
55          System.out.println("");
56          System.out.println("----------------------------------------
57          System.out.println("Student Name\t Student Age");
58          System.out.println("----------------------------------------
59          for(AGE e:arr)
60          {
61              System.out.println(e.getAge()+"\t\t"+e.getName());
62          }
63          System.out.println("----------------------------------------
64          Collections.sort( arr);
65          System.out.println("");
66          System.out.println("After sort: ");
67          System.out.println("");
68          System.out.println("----------------------------------------
69          System.out.println("Student Name\tStudent Age");
70          System.out.println("----------------------------------------
71          for(AGE e:arr)
72          {
73              System.out.println(e.getAge()+"\t\t"+e.getName());
74          }
```

```
Console
<terminated> AgeSort [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj
Before sort:

-----------------------------------------
Student Name      Student Age
-----------------------------------------
13              Sangram
21              Omkar
24              Harshad
17              Aditya
-----------------------------------------

After sort:

-----------------------------------------
Student Name    Student Age
-----------------------------------------
13              Sangram
17              Aditya
21              Omkar
24              Harshad
-----------------------------------------
```

Q.2 Create a class Book that implements Comparable and compares books based on their titles.

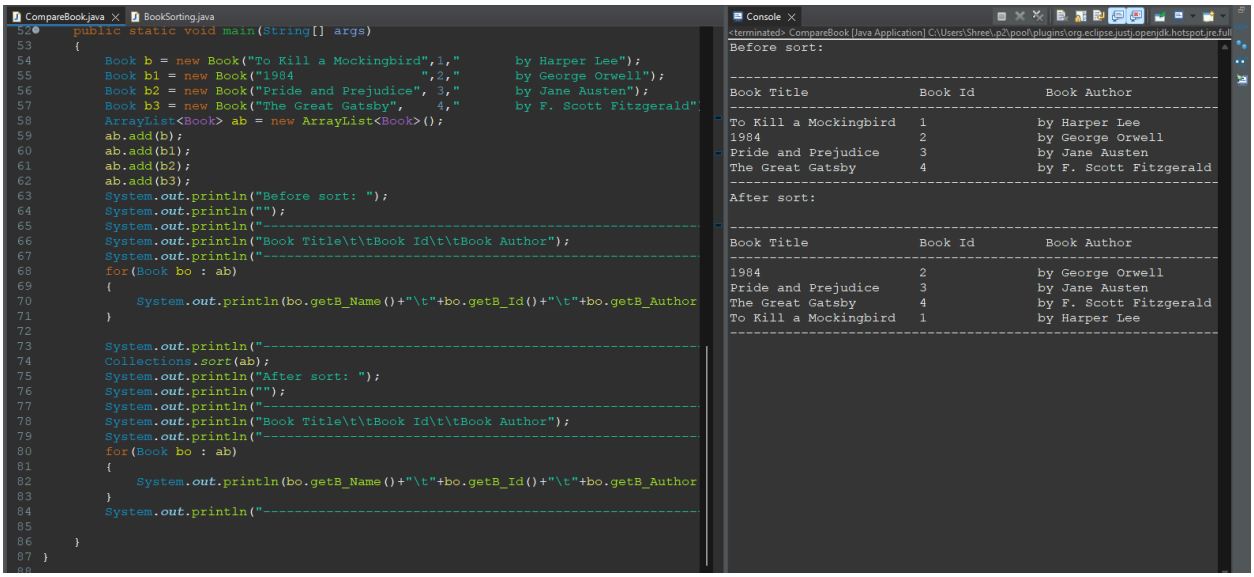
```
CompareBook.java | BookSorting.java
1  /*
2   * Q.2 Create a class Book that implements Comparable and
3   * compares books based on their titles.
4   */
5  package comparableInterface.com;
6
7  import java.util.ArrayList;
8  import java.util.Collections;
9
10 class Book implements Comparable<Book>
11 {
12     private String B_Name;
13     private int B_Id;
14     private String B_Author;
15     public Book(String b_Name, int b_Id, String b_Author)
16     {
17         this.B_Name = b_Name;
18         this.B_Id = b_Id;
19         this.B_Author = b_Author;
20     }
21     public String getB_Name()
22     {
23         return B_Name;
24     }
25     public int getB_Id()
26     {
27         return B_Id;
28     }
29     public String getB_Author()
30     {
31         return B_Author;
32     }
33     @Override
34     public int compareTo(Book o)
35     {
36         if(this.B_Name.compareTo(o.B_Name)==0)
37         {
```

```
Console
<terminated> CompareBook [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.fu
Before sort:

-----------------------------------------------------------------
Book Title              Book Id         Book Author
-----------------------------------------------------------------
To Kill a Mockingbird   1               by Harper Lee
1984                    2               by George Orwell
Pride and Prejudice     3               by Jane Austen
The Great Gatsby        4               by F. Scott Fitzgerald
-----------------------------------------------------------------
After sort:

-----------------------------------------------------------------
Book Title              Book Id         Book Author
-----------------------------------------------------------------
1984                    2               by George Orwell
Pride and Prejudice     3               by Jane Austen
The Great Gatsby        4               by F. Scott Fitzgerald
To Kill a Mockingbird   1               by Harper Lee
-----------------------------------------------------------------
```



```
CompareBook.java  BookSorting.java

        {
            return 0;
        }
        else if(this.B_Name.compareTo(o.B_Name)>0)
        {
            return 1;
        }
        else
        {
            return -1;
        }
    }
}
public class CompareBook
{
    public static void main(String[] args)
    {
        Book b = new Book("To Kill a Mockingbird",1,"       by Harper Lee");
        Book b1 = new Book("1984                  ",2,"       by George Orwell");
        Book b2 = new Book("Pride and Prejudice", 3,"       by Jane Austen");
        Book b3 = new Book("The Great Gatsby",    4,"       by F. Scott Fitzgerald");
        ArrayList<Book> ab = new ArrayList<Book>();
        ab.add(b);
        ab.add(b1);
        ab.add(b2);
        ab.add(b3);
        System.out.println("Before sort: ");
        System.out.println("");
        System.out.println("-------------------------------------------------------------
        System.out.println("Book Title\t\tBook Id\t\tBook Author");
        System.out.println("-------------------------------------------------------------
        for(Book bo : ab)
        {
            System.out.println(bo.getB_Name()+"\t"+bo.getB_Id()+"\t"+bo.getB_Author
        }

        System.out.println("-------------------------------------------------------------
```

```
Console
<terminated> CompareBook [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.fu
Before sort:

-----------------------------------------------------------------
Book Title              Book Id         Book Author
-----------------------------------------------------------------
To Kill a Mockingbird   1               by Harper Lee
1984                    2               by George Orwell
Pride and Prejudice     3               by Jane Austen
The Great Gatsby        4               by F. Scott Fitzgerald
-----------------------------------------------------------------
After sort:

-----------------------------------------------------------------
Book Title              Book Id         Book Author
-----------------------------------------------------------------
1984                    2               by George Orwell
Pride and Prejudice     3               by Jane Austen
The Great Gatsby        4               by F. Scott Fitzgerald
To Kill a Mockingbird   1               by Harper Lee
-----------------------------------------------------------------
```



```
CompareBook.java  BookSorting.java

    public static void main(String[] args)
    {
        Book b = new Book("To Kill a Mockingbird",1,"       by Harper Lee");
        Book b1 = new Book("1984                  ",2,"       by George Orwell");
        Book b2 = new Book("Pride and Prejudice", 3,"       by Jane Austen");
        Book b3 = new Book("The Great Gatsby",    4,"       by F. Scott Fitzgerald");
        ArrayList<Book> ab = new ArrayList<Book>();
        ab.add(b);
        ab.add(b1);
        ab.add(b2);
        ab.add(b3);
        System.out.println("Before sort: ");
        System.out.println("");
        System.out.println("-------------------------------------------------------------
        System.out.println("Book Title\t\tBook Id\t\tBook Author");
        System.out.println("-------------------------------------------------------------
        for(Book bo : ab)
        {
            System.out.println(bo.getB_Name()+"\t"+bo.getB_Id()+"\t"+bo.getB_Author
        }

        System.out.println("-------------------------------------------------------------
        Collections.sort(ab);
        System.out.println("After sort: ");
        System.out.println("");
        System.out.println("-------------------------------------------------------------
        System.out.println("Book Title\t\tBook Id\t\tBook Author");
        System.out.println("-------------------------------------------------------------
        for(Book bo : ab)
        {
            System.out.println(bo.getB_Name()+"\t"+bo.getB_Id()+"\t"+bo.getB_Author
        }
        System.out.println("-------------------------------------------------------------

    }
}
```

```
Console
<terminated> CompareBook [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full
Before sort:

-----------------------------------------------------------------
Book Title              Book Id         Book Author
-----------------------------------------------------------------
To Kill a Mockingbird   1               by Harper Lee
1984                    2               by George Orwell
Pride and Prejudice     3               by Jane Austen
The Great Gatsby        4               by F. Scott Fitzgerald
-----------------------------------------------------------------
After sort:

-----------------------------------------------------------------
Book Title              Book Id         Book Author
-----------------------------------------------------------------
1984                    2               by George Orwell
Pride and Prejudice     3               by Jane Austen
The Great Gatsby        4               by F. Scott Fitzgerald
To Kill a Mockingbird   1               by Harper Lee
-----------------------------------------------------------------
```

Q.3 Implement Comparable for a Product class considering their prices.

CompareBook.java | BookSorting.java | ComparePrice.java

```
import java.util.ArrayList;
import java.util.Collections;

class Product implements Comparable<Product>
{
    private int Price;
    private String Product;
    public Product(int price, String product)
    {
        this.Price = price;
        this.Product = product;
    }
    public int getPrice()
    {
        return Price;
    }
    public String getProduct()
    {
        return Product;
    }
    @Override
    public int compareTo(Product o)
    {
        if(this.Price==o.Price)
        {
            return 0;
        }
        else if(this.Price>o.Price)
        {
            return 1;
        }
        else
        {
            return -1;
        }
    }
}
```

Console

```
<terminated> ComparePrice [Java Application] C:\Users\Shree\.p2\pool\plugins\org.
Before sort:

-------------------------------------------
Product Price   Product
-------------------------------------------
123             Apple
198             Orange
342             Dragon Fruit
231             Pinapple
-------------------------------------------

After sort:

-------------------------------------------
Product Price   Product
-------------------------------------------
123             Apple
198             Orange
231             Pinapple
342             Dragon Fruit
-----------------------------
```

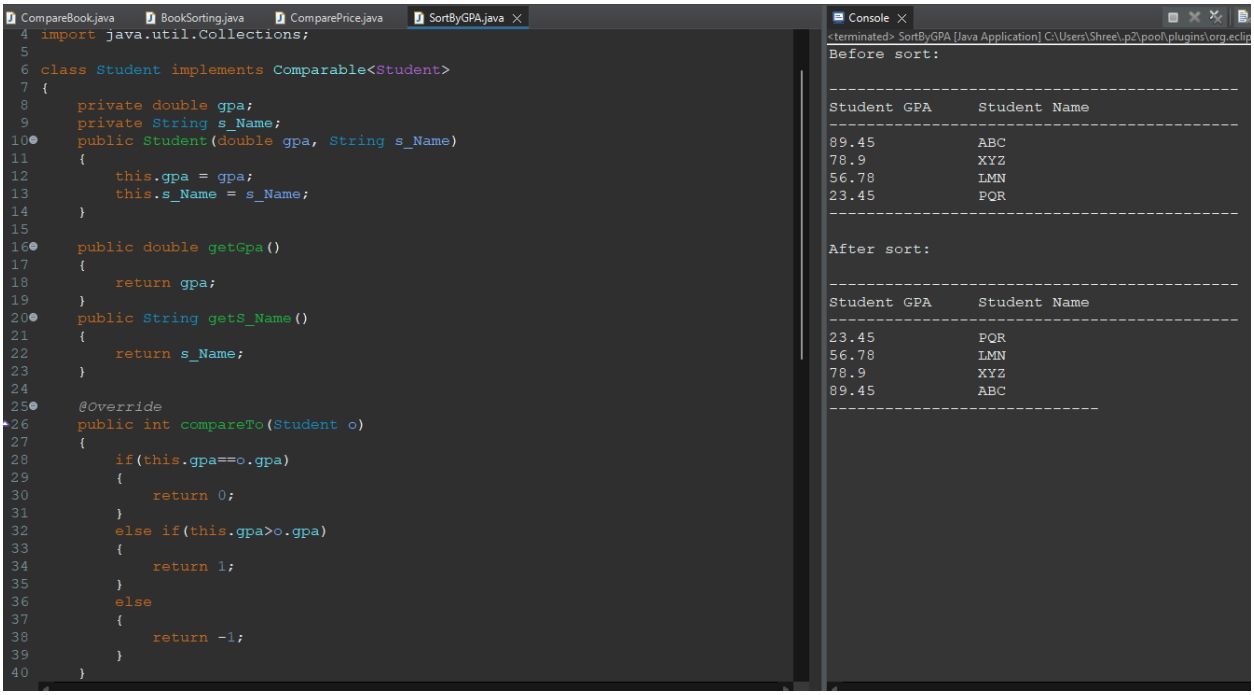
CompareBook.java | BookSorting.java | *ComparePrice.java

```
    }
}
public class ComparePrice
{
    public static void main(String[] args)
    {
        Product b = new Product(123,"Apple");
        Product b1 = new Product(198,"Orange");
        Product b2 = new Product(342,"Dragon Fruit");
        Product b3 = new Product(231,"Pinapple");
        ArrayList<Product> a1 = new ArrayList<Product>();
        a1.add(b);
        a1.add(b1);
        a1.add(b2);
        a1.add(b3);
        System.out.println("Before sort: ");
        System.out.println("");
        System.out.println("-------------------------------------------");
        System.out.println("Product Price\tProduct");
        System.out.println("-------------------------------------------");
        for(Product e:a1)
        {
            System.out.println(e.getPrice()+"\t\t"+e.getProduct());
        }
        System.out.println("-------------------------------------------");
        Collections.sort(a1);
        System.out.println("");
        System.out.println("After sort: ");
        System.out.println("");
        System.out.println("-------------------------------------------");
        System.out.println("Product Price\tProduct");
        System.out.println("-------------------------------------------");
        for(Product e:a1)
        {
            System.out.println(e.getPrice()+"\t\t"+e.getProduct());
        }
        System.out.println("-----------------------------");
```

Console

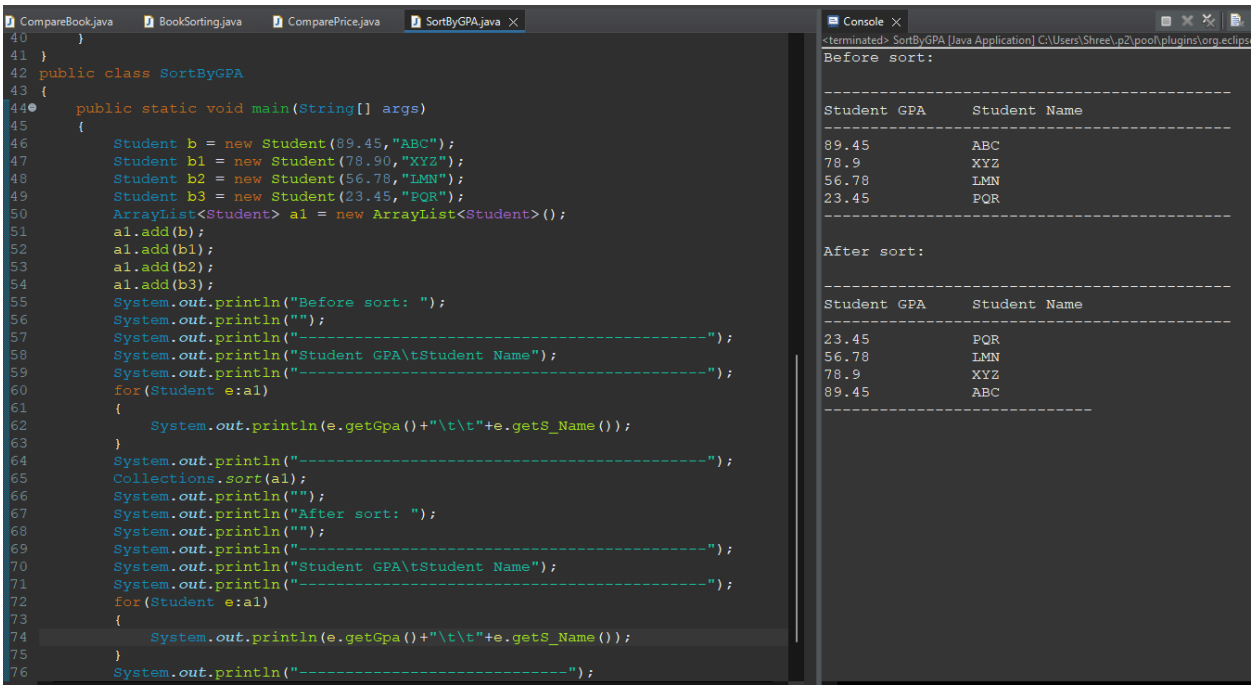
```
<terminated> ComparePrice [Java Application] C:\Users\Shree\.p2\pool\plugins\org.ecli
Before sort:

-------------------------------------------
Product Price   Product
-------------------------------------------
123             Apple
198             Orange
342             Dragon Fruit
231             Pinapple
-------------------------------------------

After sort:

-------------------------------------------
Product Price   Product
-------------------------------------------
123             Apple
198             Orange
231             Pinapple
342             Dragon Fruit
-----------------------------
```

Q.4 Make a Student class Comparable based on their GPA.



```
import java.util.Collections;

class Student implements Comparable<Student>
{
    private double gpa;
    private String s_Name;
    public Student(double gpa, String s_Name)
    {
        this.gpa = gpa;
        this.s_Name = s_Name;
    }

    public double getGpa()
    {
        return gpa;
    }
    public String getS_Name()
    {
        return s_Name;
    }

    @Override
    public int compareTo(Student o)
    {
        if(this.gpa==o.gpa)
        {
            return 0;
        }
        else if(this.gpa>o.gpa)
        {
            return 1;
        }
        else
        {
            return -1;
        }
    }
```

```
Before sort:

-------------------------------------------
Student GPA      Student Name
-------------------------------------------
89.45            ABC
78.9             XYZ
56.78            LMN
23.45            PQR
-------------------------------------------

After sort:

-------------------------------------------
Student GPA      Student Name
-------------------------------------------
23.45            PQR
56.78            LMN
78.9             XYZ
89.45            ABC
-----------------------------
```



```
}
public class SortByGPA
{
    public static void main(String[] args)
    {
        Student b = new Student(89.45,"ABC");
        Student b1 = new Student(78.90,"XYZ");
        Student b2 = new Student(56.78,"LMN");
        Student b3 = new Student(23.45,"PQR");
        ArrayList<Student> a1 = new ArrayList<Student>();
        a1.add(b);
        a1.add(b1);
        a1.add(b2);
        a1.add(b3);
        System.out.println("Before sort: ");
        System.out.println("");
        System.out.println("-------------------------------------------");
        System.out.println("Student GPA\tStudent Name");
        System.out.println("-------------------------------------------");
        for(Student e:a1)
        {
            System.out.println(e.getGpa()+"\t\t"+e.getS_Name());
        }
        System.out.println("-------------------------------------------");
        Collections.sort(a1);
        System.out.println("");
        System.out.println("After sort: ");
        System.out.println("");
        System.out.println("-------------------------------------------");
        System.out.println("Student GPA\tStudent Name");
        System.out.println("-------------------------------------------");
        for(Student e:a1)
        {
            System.out.println(e.getGpa()+"\t\t"+e.getS_Name());
        }
        System.out.println("-----------------------------");
```

```
Before sort:

-------------------------------------------
Student GPA      Student Name
-------------------------------------------
89.45            ABC
78.9             XYZ
56.78            LMN
23.45            PQR
-------------------------------------------

After sort:

-------------------------------------------
Student GPA      Student Name
-------------------------------------------
23.45            PQR
56.78            LMN
78.9             XYZ
89.45            ABC
-----------------------------
```

Q.5 Create a Song class that implements Comparable for sorting songs based on their play durations.

BookSorting.java | SortByGPA.java | PlayDuration.java

```
import java.util.Collections;

class Song implements Comparable<Song>
{
    private double Play_Time;
    private String s_Name;

    public Song(double play_Time, String s_Name)
    {
        this.Play_Time = play_Time;
        this.s_Name = s_Name;
    }

    public double getPlay_Time()
    {
        return Play_Time;
    }

    public String getS_Name()
    {
        return s_Name;
    }

    @Override
    public int compareTo(Song o)
    {
        if(this.Play_Time==o.Play_Time)
        {
            return 0;
        }
        else if(this.Play_Time>o.Play_Time)
        {
            return 1;
        }
        else
        {
            return -1;
```

Console

```
<terminated> PlayDuration [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.
Before sort:

--------------------------------------------
Song PlayTime   Song Name
--------------------------------------------
89.45           ABC
78.9            XYZ
56.78           LMN
23.45           PQR
--------------------------------------------

After sort:

--------------------------------------------
Song PlayTime   Song Name
--------------------------------------------
23.45           PQR
56.78           LMN
78.9            XYZ
89.45           ABC
--------------------------------------------
```

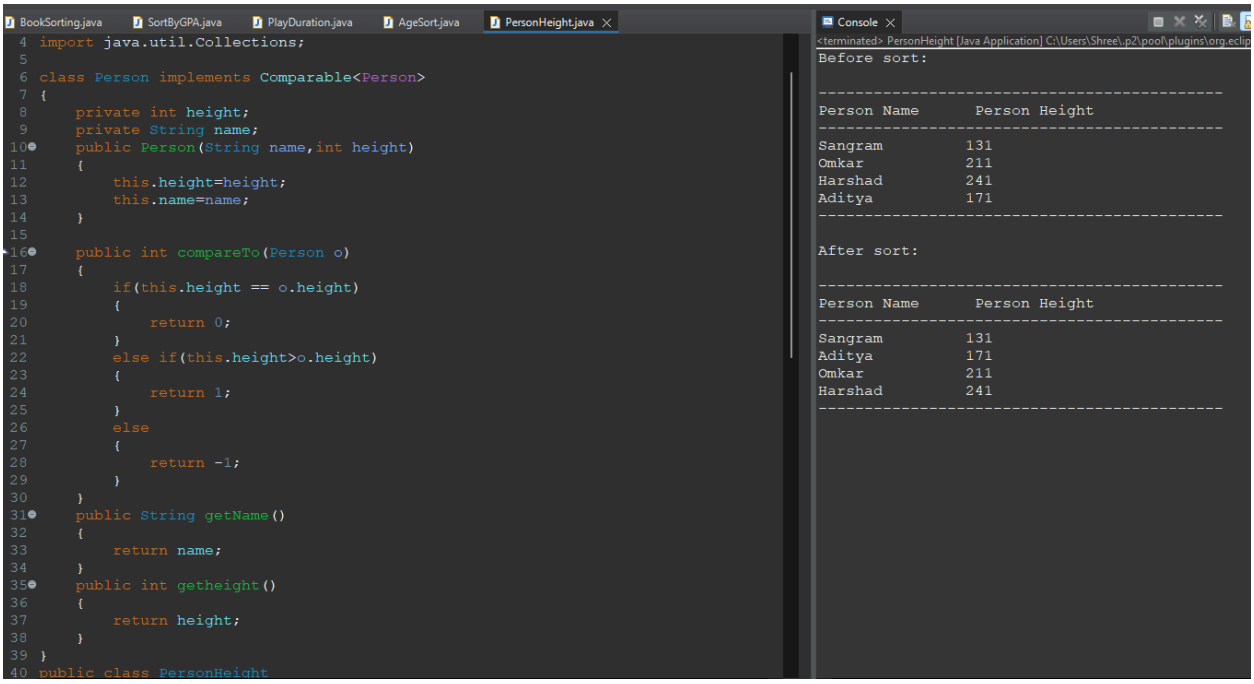
BookSorting.java | SortByGPA.java | PlayDuration.java

```
}
public class PlayDuration
{
    public static void main(String[] args)
    {
        Song b = new Song(89.45,"ABC");
        Song b1 = new Song(78.90,"XYZ");
        Song b2 = new Song(56.78,"LMN");
        Song b3 = new Song(23.45,"PQR");
        ArrayList<Song> a1 = new ArrayList<Song>();
        a1.add(b);
        a1.add(b1);
        a1.add(b2);
        a1.add(b3);
        System.out.println("Before sort: ");
        System.out.println("");
        System.out.println("--------------------------------------------");
        System.out.println("Song PlayTime\tSong Name");
        System.out.println("--------------------------------------------");
        for(Song e:a1)
        {
            System.out.println(e.getPlay_Time()+"\t\t"+e.getS_Name());
        }
        System.out.println("--------------------------------------------");
        Collections.sort(a1);
        System.out.println("");
        System.out.println("After sort: ");
        System.out.println("");
        System.out.println("--------------------------------------------");
        System.out.println("Song PlayTime\tSong Name");
        System.out.println("--------------------------------------------");
        for(Song e:a1)
        {
            System.out.println(e.getPlay_Time()+"\t\t"+e.getS_Name());
        }
        System.out.println("--------------------------------------------");
    }
```

Console

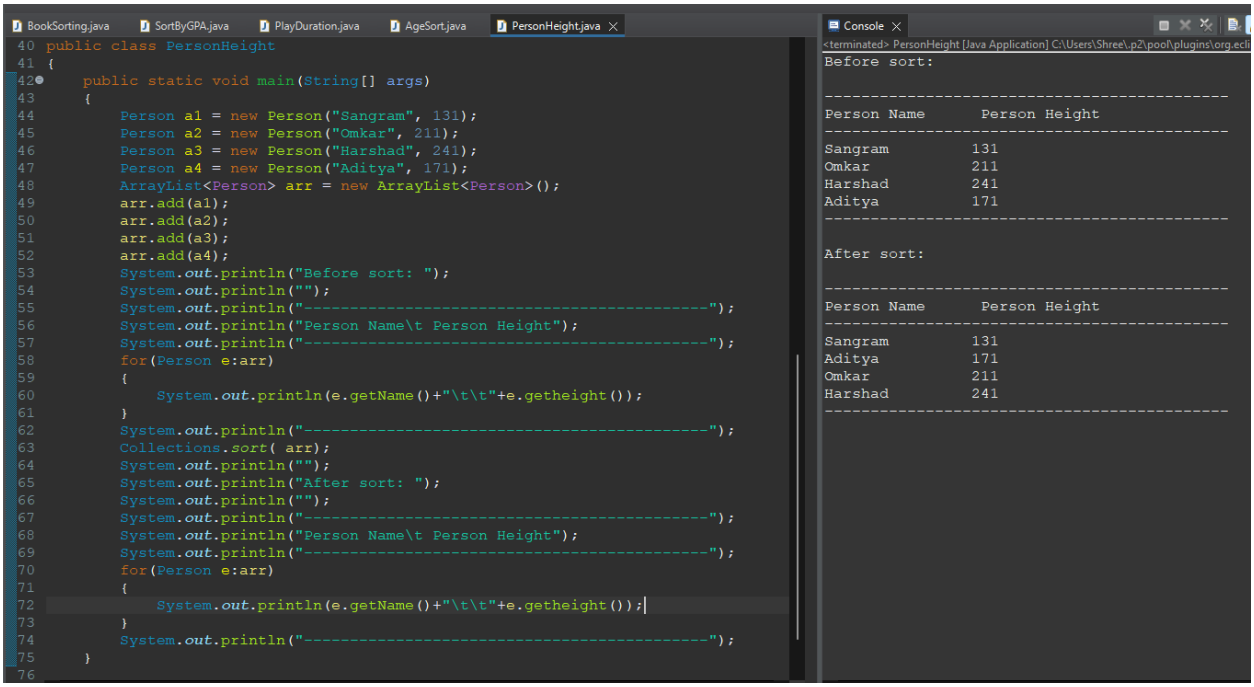
```
<terminated> PlayDuration [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse
Before sort:

--------------------------------------------
Song PlayTime   Song Name
--------------------------------------------
89.45           ABC
78.9            XYZ
56.78           LMN
23.45           PQR
--------------------------------------------

After sort:

--------------------------------------------
Song PlayTime   Song Name
--------------------------------------------
23.45           PQR
56.78           LMN
78.9            XYZ
89.45           ABC
--------------------------------------------
```

Q.6 Implement the Comparable interface for a Person class that compares instances based on their height.



```
import java.util.Collections;

class Person implements Comparable<Person>
{
    private int height;
    private String name;
    public Person(String name,int height)
    {
        this.height=height;
        this.name=name;
    }

    public int compareTo(Person o)
    {
        if(this.height == o.height)
        {
            return 0;
        }
        else if(this.height>o.height)
        {
            return 1;
        }
        else
        {
            return -1;
        }
    }
    public String getName()
    {
        return name;
    }
    public int getheight()
    {
        return height;
    }
}
public class PersonHeight
```

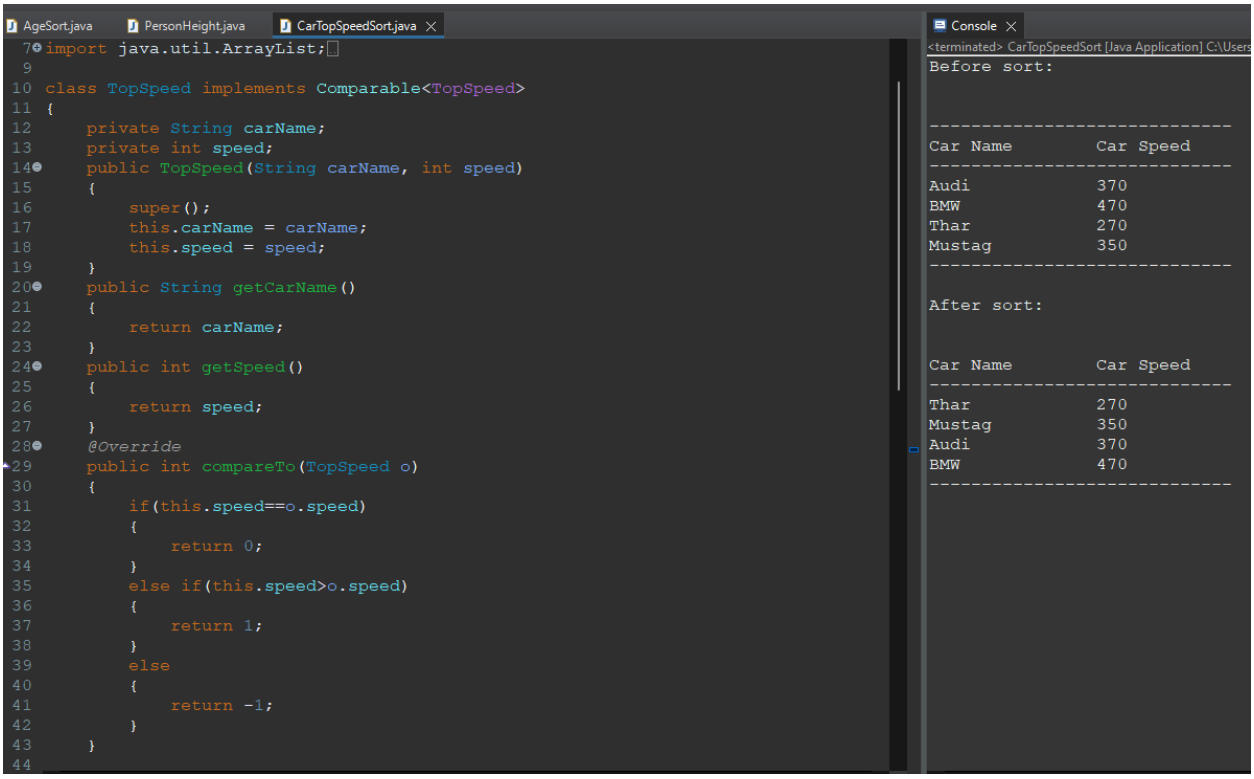
```
<terminated> PersonHeight [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclip
Before sort:

--------------------------------------------
Person Name       Person Height
--------------------------------------------
Sangram          131
Omkar            211
Harshad          241
Aditya           171
--------------------------------------------

After sort:

--------------------------------------------
Person Name       Person Height
--------------------------------------------
Sangram          131
Aditya           171
Omkar            211
Harshad          241
--------------------------------------------
```



```
public class PersonHeight
{
    public static void main(String[] args)
    {
        Person a1 = new Person("Sangram", 131);
        Person a2 = new Person("Omkar", 211);
        Person a3 = new Person("Harshad", 241);
        Person a4 = new Person("Aditya", 171);
        ArrayList<Person> arr = new ArrayList<Person>();
        arr.add(a1);
        arr.add(a2);
        arr.add(a3);
        arr.add(a4);
        System.out.println("Before sort: ");
        System.out.println("");
        System.out.println("--------------------------------------------");
        System.out.println("Person Name\t Person Height");
        System.out.println("--------------------------------------------");
        for(Person e:arr)
        {
            System.out.println(e.getName()+"\t\t"+e.getheight());
        }
        System.out.println("--------------------------------------------");
        Collections.sort( arr);
        System.out.println("");
        System.out.println("After sort: ");
        System.out.println("");
        System.out.println("--------------------------------------------");
        System.out.println("Person Name\t Person Height");
        System.out.println("--------------------------------------------");
        for(Person e:arr)
        {
            System.out.println(e.getName()+"\t\t"+e.getheight());
        }
        System.out.println("--------------------------------------------");
    }
```

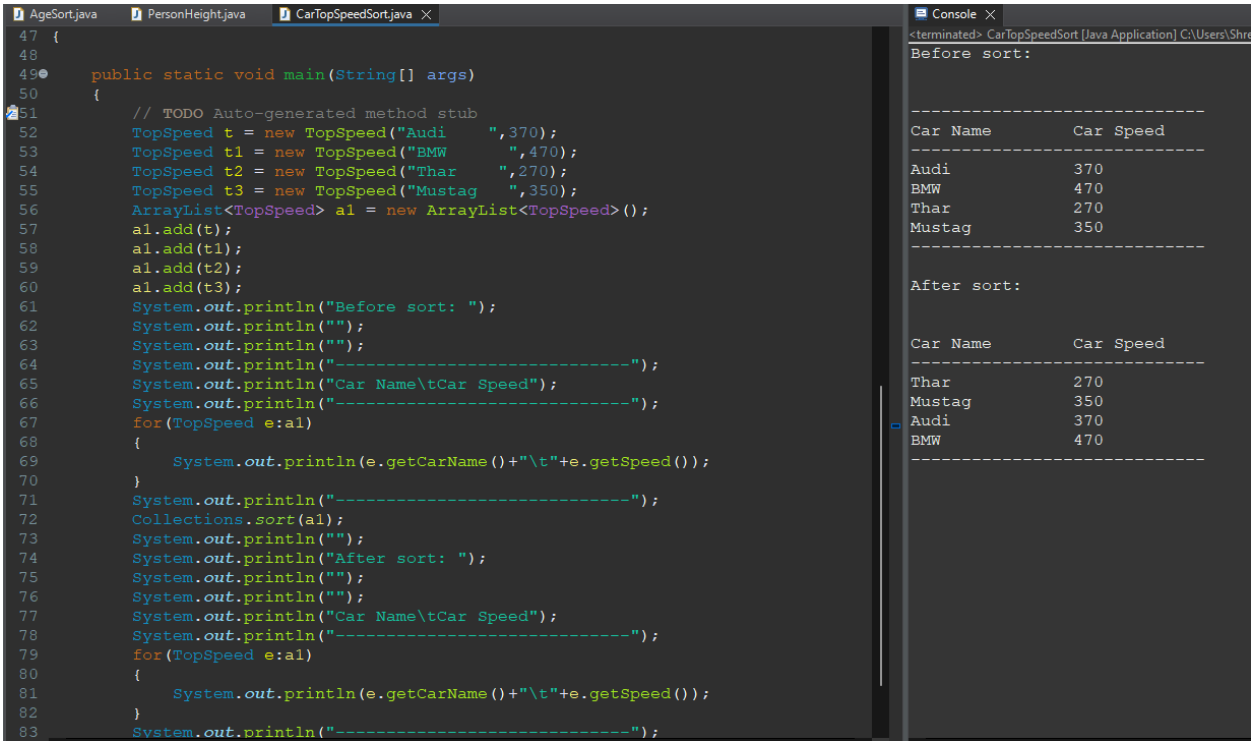
```
<terminated> PersonHeight [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclip
Before sort:

--------------------------------------------
Person Name       Person Height
--------------------------------------------
Sangram          131
Omkar            211
Harshad          241
Aditya           171
--------------------------------------------

After sort:

--------------------------------------------
Person Name       Person Height
--------------------------------------------
Sangram          131
Aditya           171
Omkar            211
Harshad          241
--------------------------------------------
```

Q.7 Create a Car class that implements Comparable and compares cars based on their top speed.



```
AgeSort.java  PersonHeight.java  CarTopSpeedSort.java

 7 import java.util.ArrayList;
 9
10 class TopSpeed implements Comparable<TopSpeed>
11 {
12     private String carName;
13     private int speed;
14     public TopSpeed(String carName, int speed)
15     {
16         super();
17         this.carName = carName;
18         this.speed = speed;
19     }
20     public String getCarName()
21     {
22         return carName;
23     }
24     public int getSpeed()
25     {
26         return speed;
27     }
28     @Override
29     public int compareTo(TopSpeed o)
30     {
31         if(this.speed==o.speed)
32         {
33             return 0;
34         }
35         else if(this.speed>o.speed)
36         {
37             return 1;
38         }
39         else
40         {
41             return -1;
42         }
43     }
44
```

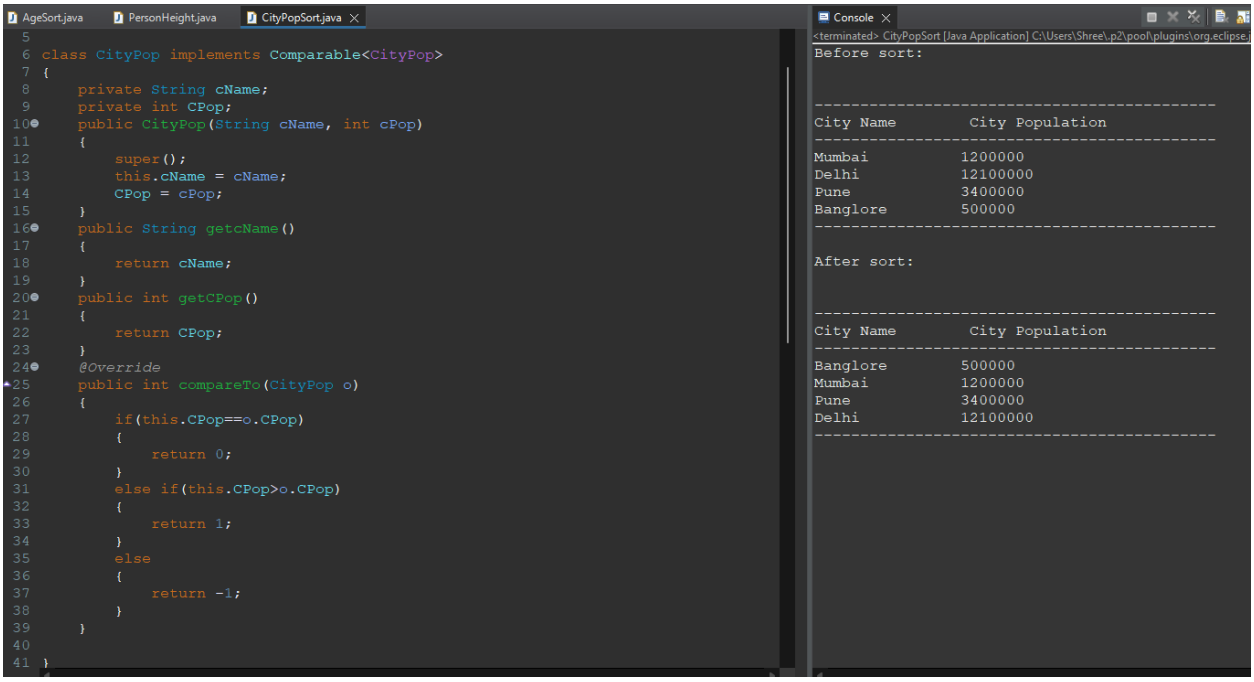
```
Console
<terminated> CarTopSpeedSort [Java Application] C:\Users
Before sort:


------------------------------
Car Name        Car Speed
------------------------------
Audi            370
BMW             470
Thar            270
Mustag          350
------------------------------

After sort:


Car Name        Car Speed
------------------------------
Thar            270
Mustag          350
Audi            370
BMW             470
------------------------------
```



```
AgeSort.java  PersonHeight.java  CarTopSpeedSort.java

47 {
48
49     public static void main(String[] args)
50     {
51         // TODO Auto-generated method stub
52         TopSpeed t = new TopSpeed("Audi     ",370);
53         TopSpeed t1 = new TopSpeed("BMW      ",470);
54         TopSpeed t2 = new TopSpeed("Thar     ",270);
55         TopSpeed t3 = new TopSpeed("Mustag   ",350);
56         ArrayList<TopSpeed> a1 = new ArrayList<TopSpeed>();
57         a1.add(t);
58         a1.add(t1);
59         a1.add(t2);
60         a1.add(t3);
61         System.out.println("Before sort: ");
62         System.out.println("");
63         System.out.println("");
64         System.out.println("------------------------------");
65         System.out.println("Car Name\tCar Speed");
66         System.out.println("------------------------------");
67         for(TopSpeed e:a1)
68         {
69             System.out.println(e.getCarName()+"\t"+e.getSpeed());
70         }
71         System.out.println("------------------------------");
72         Collections.sort(a1);
73         System.out.println("");
74         System.out.println("After sort: ");
75         System.out.println("");
76         System.out.println("");
77         System.out.println("Car Name\tCar Speed");
78         System.out.println("------------------------------");
79         for(TopSpeed e:a1)
80         {
81             System.out.println(e.getCarName()+"\t"+e.getSpeed());
82         }
83         System.out.println("------------------------------");
```

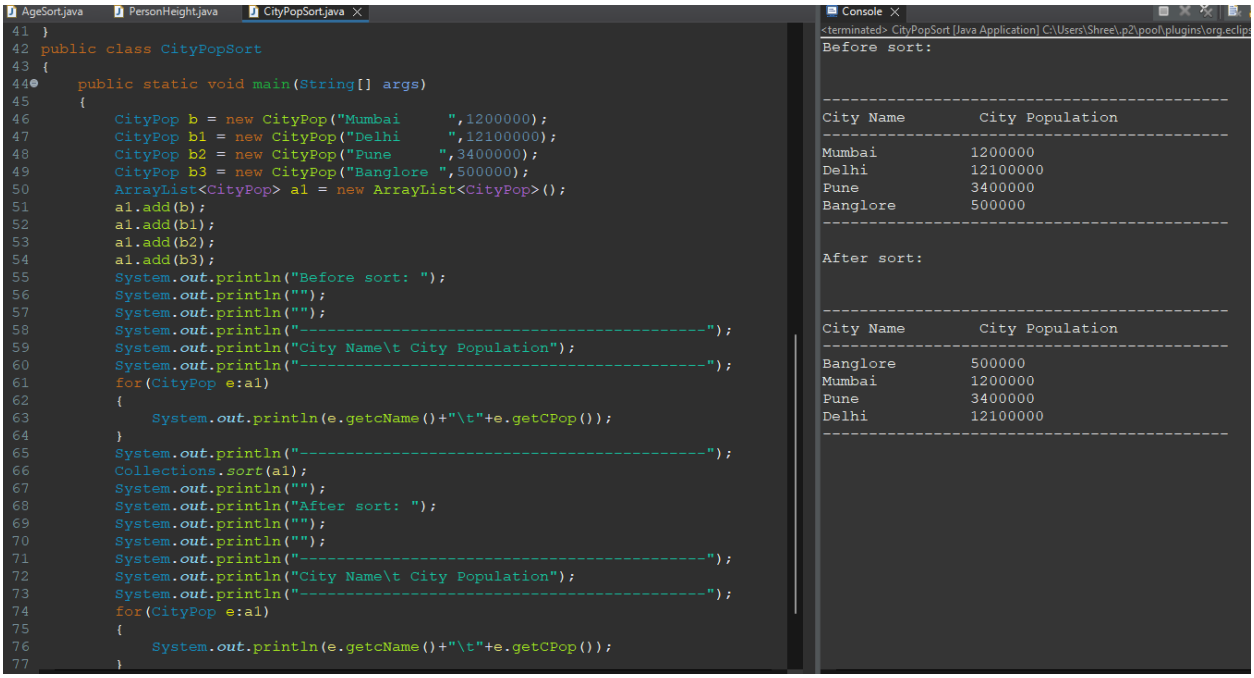
```
Console
<terminated> CarTopSpeedSort [Java Application] C:\Users\Shre
Before sort:


------------------------------
Car Name        Car Speed
------------------------------
Audi            370
BMW             470
Thar            270
Mustag          350
------------------------------

After sort:


Car Name        Car Speed
------------------------------
Thar            270
Mustag          350
Audi            370
BMW             470
------------------------------
```

Q.8 Implement Comparable for a Country class considering their population.



```
AgeSort.java  PersonHeight.java  CityPopSort.java

class CityPop implements Comparable<CityPop>
{
    private String cName;
    private int CPop;
    public CityPop(String cName, int cPop)
    {
        super();
        this.cName = cName;
        CPop = cPop;
    }
    public String getcName()
    {
        return cName;
    }
    public int getCPop()
    {
        return CPop;
    }
    @Override
    public int compareTo(CityPop o)
    {
        if(this.CPop==o.CPop)
        {
            return 0;
        }
        else if(this.CPop>o.CPop)
        {
            return 1;
        }
        else
        {
            return -1;
        }
    }

}
```

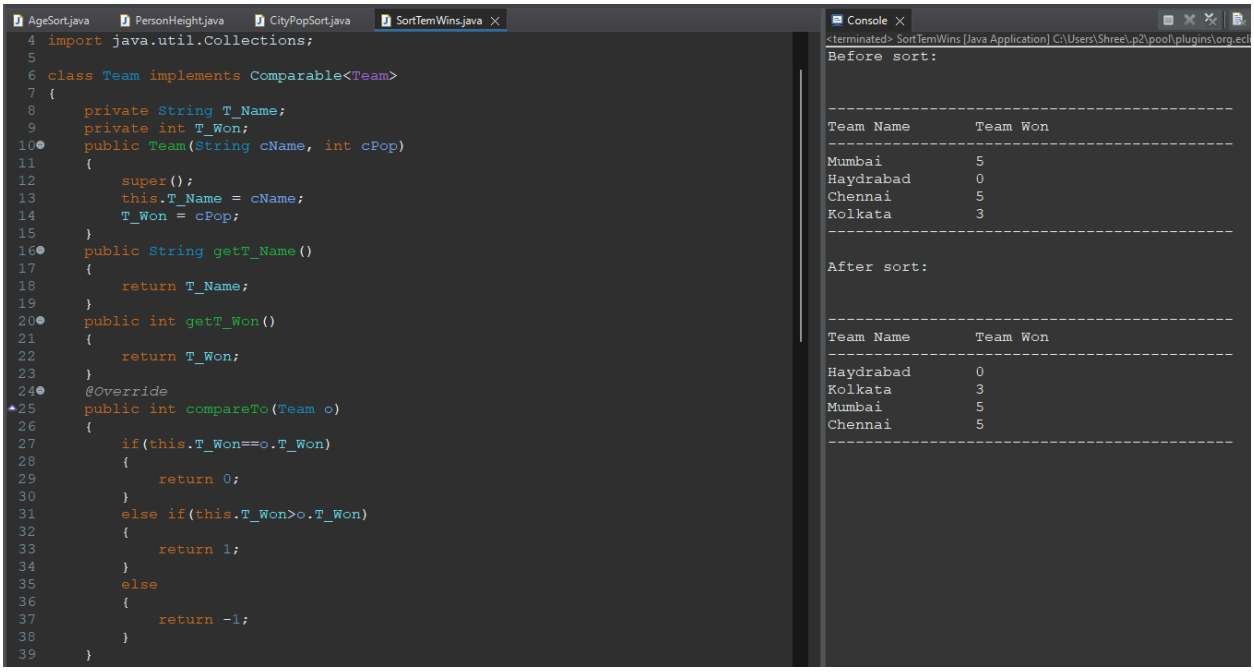
```
Console
<terminated> CityPopSort [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.j
Before sort:


-------------------------------------------
City Name         City Population
-------------------------------------------
Mumbai           1200000
Delhi            12100000
Pune             3400000
Banglore         500000
-------------------------------------------

After sort:


-------------------------------------------
City Name         City Population
-------------------------------------------
Banglore         500000
Mumbai           1200000
Pune             3400000
Delhi            12100000
-------------------------------------------
```



```
AgeSort.java  PersonHeight.java  CityPopSort.java

}
public class CityPopSort
{
    public static void main(String[] args)
    {
        CityPop b = new CityPop("Mumbai     ",1200000);
        CityPop b1 = new CityPop("Delhi     ",12100000);
        CityPop b2 = new CityPop("Pune      ",3400000);
        CityPop b3 = new CityPop("Banglore ",500000);
        ArrayList<CityPop> a1 = new ArrayList<CityPop>();
        a1.add(b);
        a1.add(b1);
        a1.add(b2);
        a1.add(b3);
        System.out.println("Before sort: ");
        System.out.println("");
        System.out.println("");
        System.out.println("-------------------------------------------");
        System.out.println("City Name\t City Population");
        System.out.println("-------------------------------------------");
        for(CityPop e:a1)
        {
            System.out.println(e.getcName()+"\t"+e.getCPop());
        }
        System.out.println("-------------------------------------------");
        Collections.sort(a1);
        System.out.println("");
        System.out.println("After sort: ");
        System.out.println("");
        System.out.println("");
        System.out.println("-------------------------------------------");
        System.out.println("City Name\t City Population");
        System.out.println("-------------------------------------------");
        for(CityPop e:a1)
        {
            System.out.println(e.getcName()+"\t"+e.getCPop());
        }
```

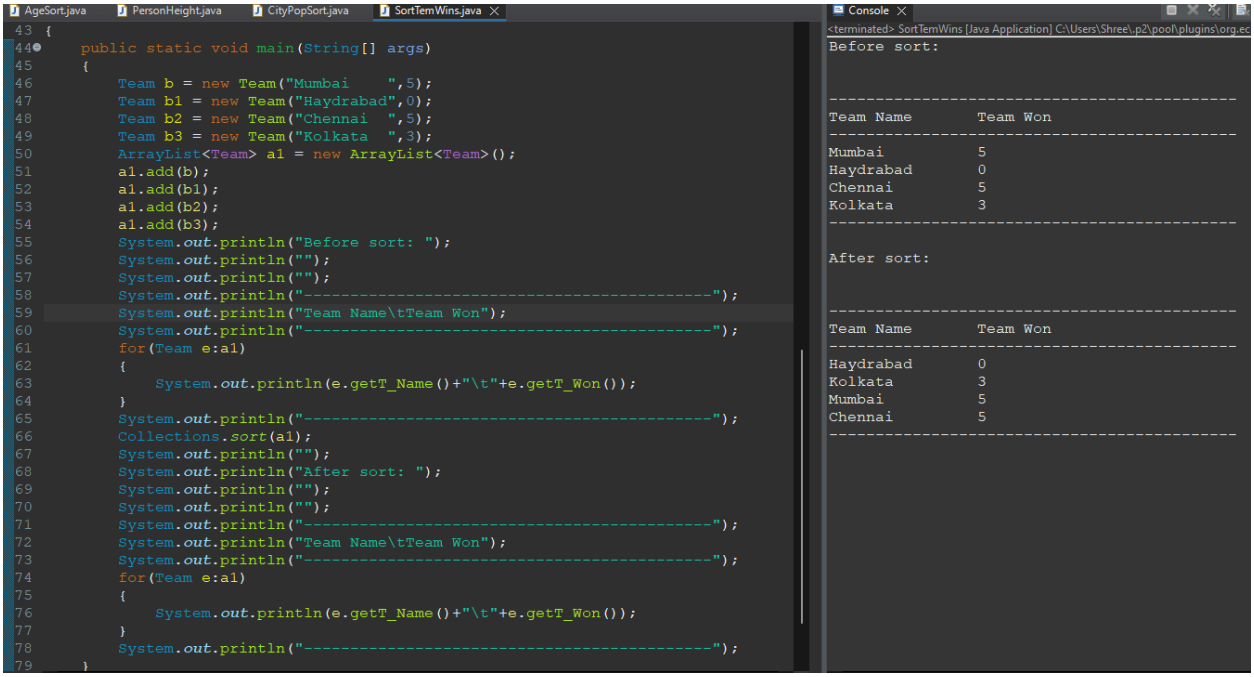
```
Console
<terminated> CityPopSort [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclips
Before sort:


-------------------------------------------
City Name         City Population
-------------------------------------------
Mumbai           1200000
Delhi            12100000
Pune             3400000
Banglore         500000
-------------------------------------------

After sort:


-------------------------------------------
City Name         City Population
-------------------------------------------
Banglore         500000
Mumbai           1200000
Pune             3400000
Delhi            12100000
-------------------------------------------
```

Q.9 Make a Team class Comparable based on the number of games won.



```
import java.util.Collections;

class Team implements Comparable<Team>
{
    private String T_Name;
    private int T_Won;
    public Team(String cName, int cPop)
    {
        super();
        this.T_Name = cName;
        T_Won = cPop;
    }
    public String getT_Name()
    {
        return T_Name;
    }
    public int getT_Won()
    {
        return T_Won;
    }
    @Override
    public int compareTo(Team o)
    {
        if(this.T_Won==o.T_Won)
        {
            return 0;
        }
        else if(this.T_Won>o.T_Won)
        {
            return 1;
        }
        else
        {
            return -1;
        }
    }
```

```
Before sort:


-------------------------------------------
Team Name       Team Won
-------------------------------------------
Mumbai          5
Haydrabad       0
Chennai         5
Kolkata         3
-------------------------------------------

After sort:


-------------------------------------------
Team Name       Team Won
-------------------------------------------
Haydrabad       0
Kolkata         3
Mumbai          5
Chennai         5
-------------------------------------------
```



```
{
    public static void main(String[] args)
    {
        Team b = new Team("Mumbai    ",5);
        Team b1 = new Team("Haydrabad",0);
        Team b2 = new Team("Chennai  ",5);
        Team b3 = new Team("Kolkata  ",3);
        ArrayList<Team> a1 = new ArrayList<Team>();
        a1.add(b);
        a1.add(b1);
        a1.add(b2);
        a1.add(b3);
        System.out.println("Before sort: ");
        System.out.println("");
        System.out.println("");
        System.out.println("-------------------------------------------");
        System.out.println("Team Name\tTeam Won");
        System.out.println("-------------------------------------------");
        for(Team e:a1)
        {
            System.out.println(e.getT_Name()+"\t"+e.getT_Won());
        }
        System.out.println("-------------------------------------------");
        Collections.sort(a1);
        System.out.println("");
        System.out.println("After sort: ");
        System.out.println("");
        System.out.println("");
        System.out.println("-------------------------------------------");
        System.out.println("Team Name\tTeam Won");
        System.out.println("-------------------------------------------");
        for(Team e:a1)
        {
            System.out.println(e.getT_Name()+"\t"+e.getT_Won());
        }
        System.out.println("-------------------------------------------");
    }
```

```
Before sort:


-------------------------------------------
Team Name       Team Won
-------------------------------------------
Mumbai          5
Haydrabad       0
Chennai         5
Kolkata         3
-------------------------------------------

After sort:


-------------------------------------------
Team Name       Team Won
-------------------------------------------
Haydrabad       0
Kolkata         3
Mumbai          5
Chennai         5
-------------------------------------------
```

Q.10 Design a City class that implements Comparable for sorting cities based on their population density.

AgeSort.java | PersonHeight.java | CityPopSort.java | SortTemWins.java

```
1  package comparable.interfsce.com;
2
3  import java.util.ArrayList;
5
6  class CityPop implements Comparable<CityPop>
7  {
8      private String cName;
9      private int CPop;
10     public CityPop(String cName, int cPop)
11     {
12         super();
13         this.cName = cName;
14         CPop = cPop;
15     }
16     public String getcName()
17     {
18         return cName;
19     }
20     public int getCPop()
21     {
22         return CPop;
23     }
24     @Override
25     public int compareTo(CityPop o)
26     {
27         if(this.CPop==o.CPop)
28         {
29             return 0;
30         }
31         else if(this.CPop>o.CPop)
32         {
33             return 1;
34         }
35         else
36         {
37             return -1;
38         }
```

Console

```
<terminated> CityPopSort [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.j
Before sort:


--------------------------------------------
City Name        City Population
--------------------------------------------
Mumbai           1200000
Delhi            12100000
Pune             3400000
Banglore         500000
--------------------------------------------

After sort:


--------------------------------------------
City Name        City Population
--------------------------------------------
Banglore         500000
Mumbai           1200000
Pune             3400000
Delhi            12100000
--------------------------------------------
```

AgeSort.java | PersonHeight.java | CityPopSort.java | SortTemWins.java

```
41 }
42 public class CityPopSort
43 {
44     public static void main(String[] args)
45     {
46         CityPop b = new CityPop("Mumbai     ",1200000);
47         CityPop b1 = new CityPop("Delhi     ",12100000);
48         CityPop b2 = new CityPop("Pune      ",3400000);
49         CityPop b3 = new CityPop("Banglore ",500000);
50         ArrayList<CityPop> a1 = new ArrayList<CityPop>();
51         a1.add(b);
52         a1.add(b1);
53         a1.add(b2);
54         a1.add(b3);
55         System.out.println("Before sort: ");
56         System.out.println("");
57         System.out.println("");
58         System.out.println("--------------------------------------------");
59         System.out.println("City Name\t City Population");
60         System.out.println("--------------------------------------------");
61         for(CityPop e:a1)
62         {
63             System.out.println(e.getcName()+"\t"+e.getCPop());
64         }
65         System.out.println("--------------------------------------------");
66         Collections.sort(a1);
67         System.out.println("");
68         System.out.println("After sort: ");
69         System.out.println("");
70         System.out.println("");
71         System.out.println("--------------------------------------------");
72         System.out.println("City Name\t City Population");
73         System.out.println("--------------------------------------------");
74         for(CityPop e:a1)
75         {
76             System.out.println(e.getcName()+"\t"+e.getCPop());
77         }
```

Console

```
<terminated> CityPopSort [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclips
Before sort:


--------------------------------------------
City Name        City Population
--------------------------------------------
Mumbai           1200000
Delhi            12100000
Pune             3400000
Banglore         500000
--------------------------------------------

After sort:


--------------------------------------------
City Name        City Population
--------------------------------------------
Banglore         500000
Mumbai           1200000
Pune             3400000
Delhi            12100000
--------------------------------------------
```

Comparator:

Q.11 Write a Comparator to sort Employee objects by their salary in descending order.

AgeSort.java | SortSalary.java | ChaininComparatorOfCarAttributes.java

```
class ESalary
{
    private int salary;
    private String name;
    public ESalary(int salary, String name)
    {
        this.salary = salary;
        this.name = name;
    }
    public int getSalary()
    {
        return salary;
    }
    public String getName()
    {
        return name;
    }
}
class Emp_Salary implements Comparator<ESalary>
{

    @Override
    public int compare(ESalary o1, ESalary o2)
    {
        if(o1.getSalary()==o2.getSalary())
        {
            return 0;
        }
        else if(o1.getSalary()>o2.getSalary())
        {
            return 1;
        }
        else
        {
            return -1;
        }
    }
```

Console

```
<terminated> SortSalary [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.o
Before sort:

-----------------------------------------
Employee Salary Employee Name
-----------------------------------------
90000           Sangram
50000           Harshad
80000           Omkar
70000           Aditya
-----------------------------------------

After sort:

-----------------------------------------
Employee Salary Employee Name
-----------------------------------------
50000           Harshad
70000           Aditya
80000           Omkar
90000           Sangram
-----------------------------------------
```

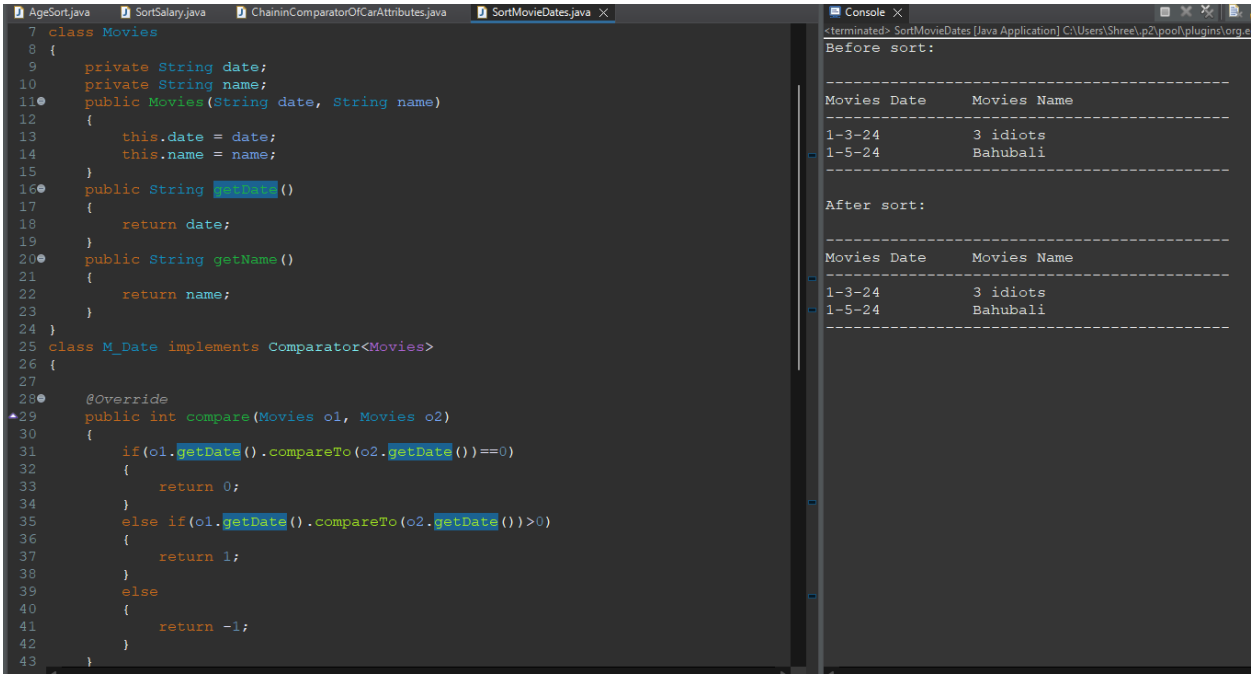
AgeSort.java | SortSalary.java | ChaininComparatorOfCarAttributes.java

```
}
public class SortSalary
{
    public static void main(String[] args)
    {
        ArrayList<ESalary> arr = new ArrayList<ESalary>();
        arr.add(new ESalary(90000, "Sangram"));
        arr.add(new ESalary(50000, "Harshad"));
        arr.add(new ESalary(80000, "Omkar"));
        arr.add(new ESalary(70000, "Aditya"));
        System.out.println("Before sort: ");
        System.out.println("");
        System.out.println("-----------------------------------------");
        System.out.println("Employee Salary\tEmployee Name");
        System.out.println("-----------------------------------------");
        for(ESalary e:arr)
        {
            System.out.println(e.getSalary()+"\t\t"+e.getName());
        }
        System.out.println("-----------------------------------------");
        Collections.sort(arr, new Emp_Salary());
        System.out.println("");
        System.out.println("After sort: ");
        System.out.println("");
        System.out.println("-----------------------------------------");
        System.out.println("Employee Salary\tEmployee Name");
        System.out.println("-----------------------------------------");
        for(ESalary e:arr)
        {
            System.out.println(e.getSalary()+"\t\t"+e.getName());
        }
        System.out.println("-----------------------------------------");
    }

}
```

Console

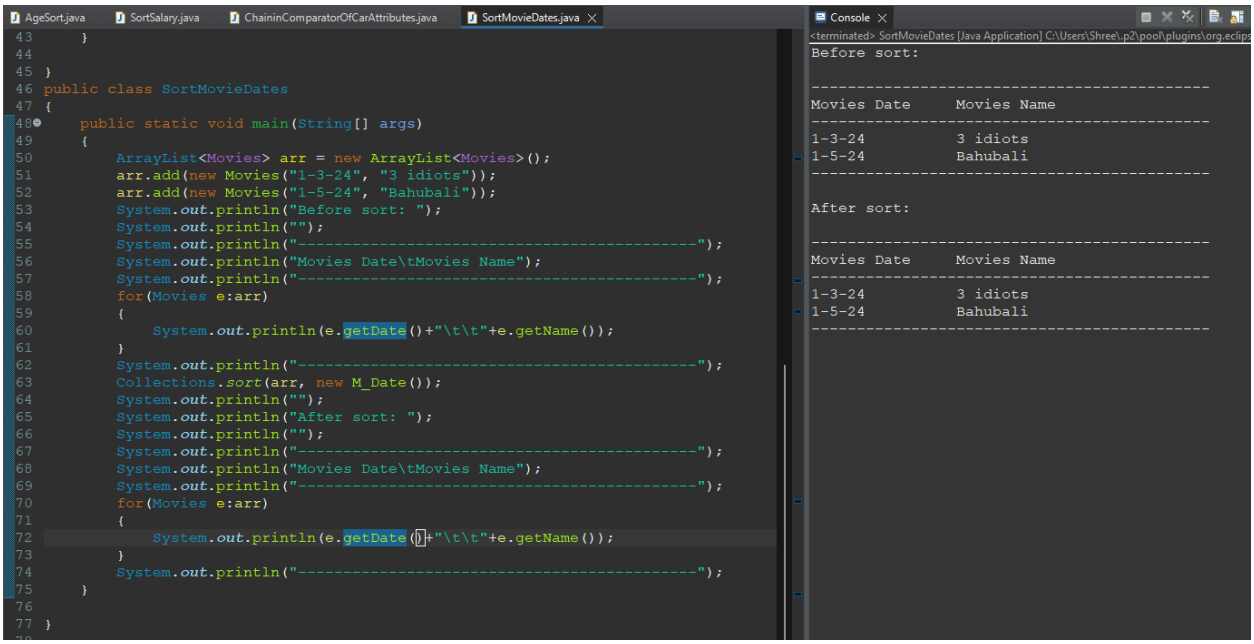
```
<terminated> SortSalary [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.ju
Before sort:

-----------------------------------------
Employee Salary Employee Name
-----------------------------------------
90000           Sangram
50000           Harshad
80000           Omkar
70000           Aditya
-----------------------------------------

After sort:

-----------------------------------------
Employee Salary Employee Name
-----------------------------------------
50000           Harshad
70000           Aditya
80000           Omkar
90000           Sangram
-----------------------------------------
```

Q.12 Create a Movie class and implement a Comparator to sort movies by release year.



```
AgeSort.java  SortSalary.java  ChaininComparatorOfCarAttributes.java  SortMovieDates.java

 7 class Movies
 8 {
 9     private String date;
10     private String name;
11     public Movies(String date, String name)
12     {
13         this.date = date;
14         this.name = name;
15     }
16     public String getDate()
17     {
18         return date;
19     }
20     public String getName()
21     {
22         return name;
23     }
24 }
25 class M_Date implements Comparator<Movies>
26 {
27
28     @Override
29     public int compare(Movies o1, Movies o2)
30     {
31         if(o1.getDate().compareTo(o2.getDate())==0)
32         {
33             return 0;
34         }
35         else if(o1.getDate().compareTo(o2.getDate())>0)
36         {
37             return 1;
38         }
39         else
40         {
41             return -1;
42         }
43     }
```

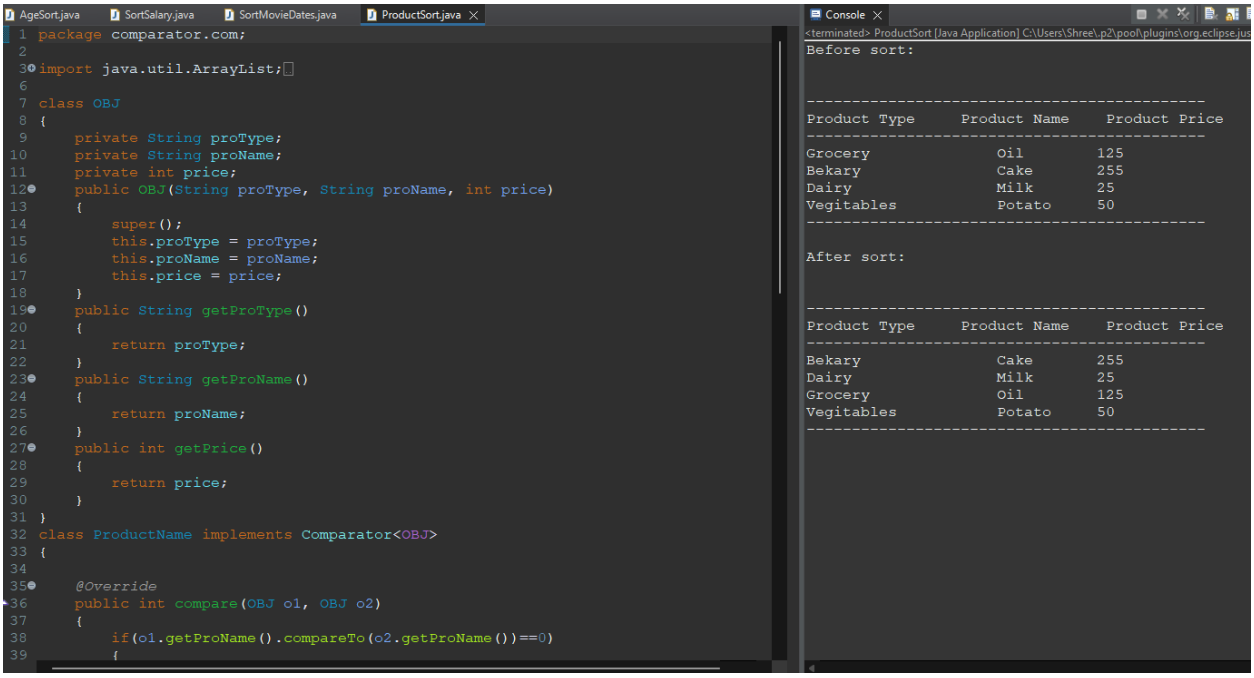
```
Console
<terminated> SortMovieDates [Java Application] C:\Users\Shree\.p2\pool\plugins\org.e
Before sort:

-------------------------------------------
Movies Date     Movies Name
-------------------------------------------
1-3-24          3 idiots
1-5-24          Bahubali
-------------------------------------------

After sort:

-------------------------------------------
Movies Date     Movies Name
-------------------------------------------
1-3-24          3 idiots
1-5-24          Bahubali
-------------------------------------------
```



```
AgeSort.java  SortSalary.java  ChaininComparatorOfCarAttributes.java  SortMovieDates.java

43     }
44
45 }
46 public class SortMovieDates
47 {
48     public static void main(String[] args)
49     {
50         ArrayList<Movies> arr = new ArrayList<Movies>();
51         arr.add(new Movies("1-3-24", "3 idiots"));
52         arr.add(new Movies("1-5-24", "Bahubali"));
53         System.out.println("Before sort: ");
54         System.out.println("");
55         System.out.println("-------------------------------------------");
56         System.out.println("Movies Date\tMovies Name");
57         System.out.println("-------------------------------------------");
58         for(Movies e:arr)
59         {
60             System.out.println(e.getDate()+"\t\t"+e.getName());
61         }
62         System.out.println("-------------------------------------------");
63         Collections.sort(arr, new M_Date());
64         System.out.println("");
65         System.out.println("After sort: ");
66         System.out.println("");
67         System.out.println("-------------------------------------------");
68         System.out.println("Movies Date\tMovies Name");
69         System.out.println("-------------------------------------------");
70         for(Movies e:arr)
71         {
72             System.out.println(e.getDate()+"\t\t"+e.getName());
73         }
74         System.out.println("-------------------------------------------");
75     }
76
77 }
```

```
Console
<terminated> SortMovieDates [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclips
Before sort:

-------------------------------------------
Movies Date     Movies Name
-------------------------------------------
1-3-24          3 idiots
1-5-24          Bahubali
-------------------------------------------

After sort:

-------------------------------------------
Movies Date     Movies Name
-------------------------------------------
1-3-24          3 idiots
1-5-24          Bahubali
-------------------------------------------
```

Q.13 Implement a Comparator to sort Product objects by name in ascending order.



```
AgeSort.java  SortSalary.java  SortMovieDates.java  ProductSort.java

 1 package comparator.com;
 2
 3 import java.util.ArrayList;
 6
 7 class OBJ
 8 {
 9     private String proType;
10     private String proName;
11     private int price;
12     public OBJ(String proType, String proName, int price)
13     {
14         super();
15         this.proType = proType;
16         this.proName = proName;
17         this.price = price;
18     }
19     public String getProType()
20     {
21         return proType;
22     }
23     public String getProName()
24     {
25         return proName;
26     }
27     public int getPrice()
28     {
29         return price;
30     }
31 }
32 class ProductName implements Comparator<OBJ>
33 {
34
35     @Override
36     public int compare(OBJ o1, OBJ o2)
37     {
38         if(o1.getProName().compareTo(o2.getProName())==0)
39         {
```

```
Console
<terminated> ProductSort [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.jus
Before sort:


-------------------------------------------
Product Type      Product Name     Product Price
-------------------------------------------
Grocery           Oil              125
Bekary            Cake             255
Dairy             Milk             25
Vegitables        Potato           50
-------------------------------------------

After sort:


-------------------------------------------
Product Type      Product Name     Product Price
-------------------------------------------
Bekary            Cake             255
Dairy             Milk             25
Grocery           Oil              125
Vegitables        Potato           50
-------------------------------------------
```

AgeSort.java | SortSalary.java | SortMovieDates.java | ProductSort.java

```
    public int compare(OBJ o1, OBJ o2)
    {
        if(o1.getProName().compareTo(o2.getProName())==0)
        {
            return 0;
        }
        else if(o1.getProName().compareTo(o2.getProName())>0)
        {
            return 1;
        }
        else
        {
            return -1;
        }
    }

}
public class ProductSort
{

    public static void main(String[] args)
    {
        ArrayList<OBJ> a1 = new ArrayList<OBJ>();
        a1.add(new OBJ("Grocery     ","     Oil    ",125));
        a1.add(new OBJ("Bekary      ","     Cake   ",255));
        a1.add(new OBJ("Dairy       ","     Milk   ",25));
        a1.add(new OBJ("Vegitables ","     Potato ",50));
        System.out.println("Before sort: ");
        System.out.println("");
        System.out.println("");
        System.out.println("-----------------------------------------");
        System.out.println("Product Type\t Product Name\t Product Price");
        System.out.println("-----------------------------------------");
        for(OBJ e:a1)
        {
            System.out.println(e.getProType()+"\t"+e.getProName()+"\t"+e.getPrice
        }
```

Console

```
<terminated> ProductSort [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj
Before sort:


-----------------------------------------
Product Type      Product Name     Product Price
-----------------------------------------
Grocery              Oil         125
Bekary               Cake        255
Dairy                Milk        25
Vegitables           Potato      50
-----------------------------------------

After sort:


-----------------------------------------
Product Type      Product Name     Product Price
-----------------------------------------
Bekary               Cake        255
Dairy                Milk        25
Grocery              Oil         125
Vegitables           Potato      50
-----------------------------------------
```

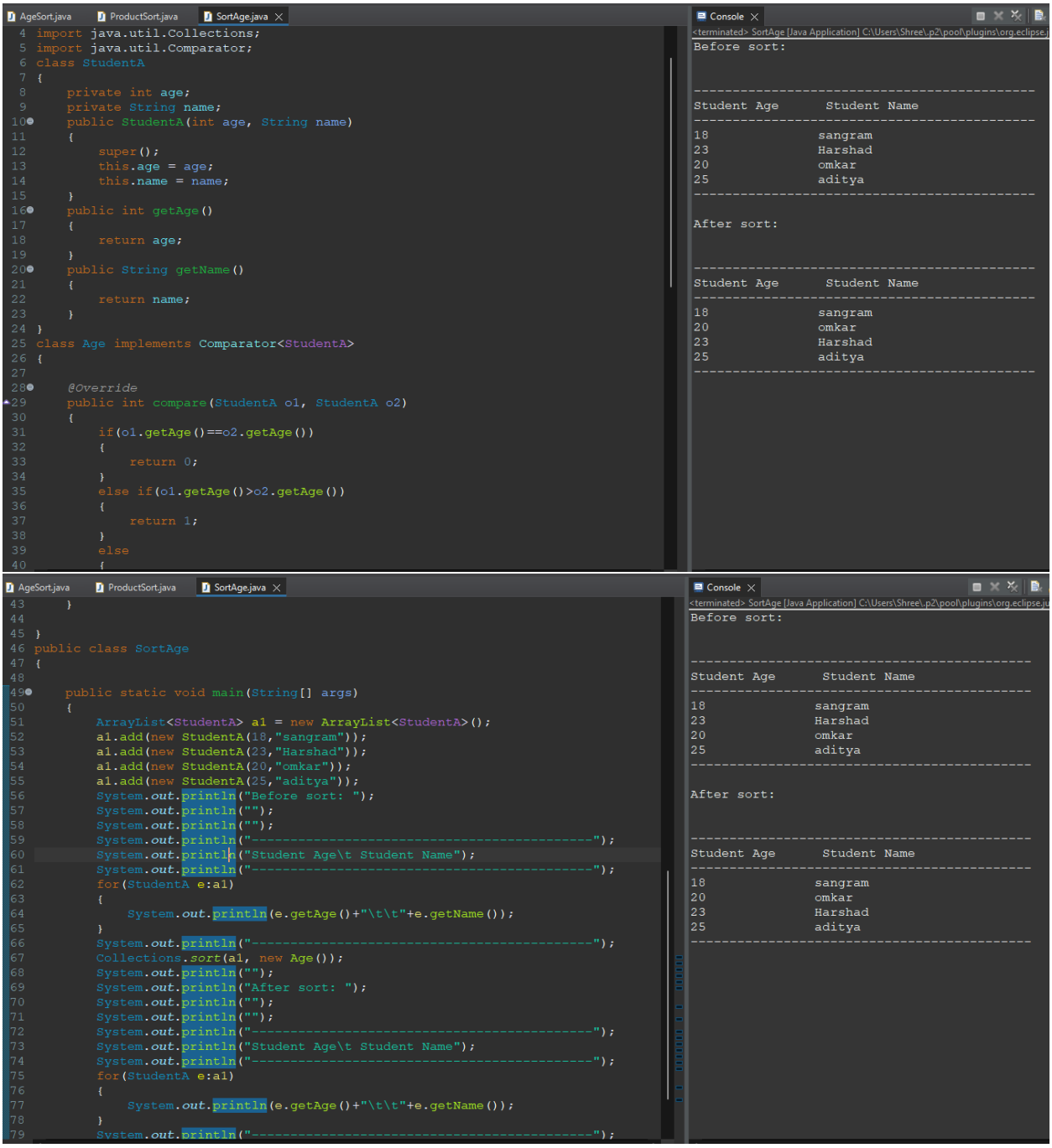
AgeSort.java | SortSalary.java | SortMovieDates.java | ProductSort.java

```
{

    public static void main(String[] args)
    {
        ArrayList<OBJ> a1 = new ArrayList<OBJ>();
        a1.add(new OBJ("Grocery     ","     Oil    ",125));
        a1.add(new OBJ("Bekary      ","     Cake   ",255));
        a1.add(new OBJ("Dairy       ","     Milk   ",25));
        a1.add(new OBJ("Vegitables ","     Potato ",50));
        System.out.println("Before sort: ");
        System.out.println("");
        System.out.println("");
        System.out.println("-----------------------------------------");
        System.out.println("Product Type\t Product Name\t Product Price");
        System.out.println("-----------------------------------------");
        for(OBJ e:a1)
        {
            System.out.println(e.getProType()+"\t"+e.getProName()+"\t"+e.getPrice
        }
        System.out.println("-----------------------------------------");
        Collections.sort(a1, new ProductName());
        System.out.println("");
        System.out.println("After sort: ");
        System.out.println("");
        System.out.println("");
        System.out.println("-----------------------------------------");
        System.out.println("Product Type\t Product Name\t Product Price");
        System.out.println("-----------------------------------------");
        for(OBJ e:a1)
        {
            System.out.println(e.getProType()+"\t"+e.getProName()+"\t"+e.getPrice
        }
        System.out.println("-----------------------------------------");
    }

}
```

Console

```
<terminated> ProductSort [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj.ope
Before sort:


-----------------------------------------
Product Type      Product Name     Product Price
-----------------------------------------
Grocery              Oil         125
Bekary               Cake        255
Dairy                Milk        25
Vegitables           Potato      50
-----------------------------------------

After sort:


-----------------------------------------
Product Type      Product Name     Product Price
-----------------------------------------
Bekary               Cake        255
Dairy                Milk        25
Grocery              Oil         125
Vegitables           Potato      50
-----------------------------------------
```

Q. 14 Write a Comparator to sort a list of Student objects by their age.



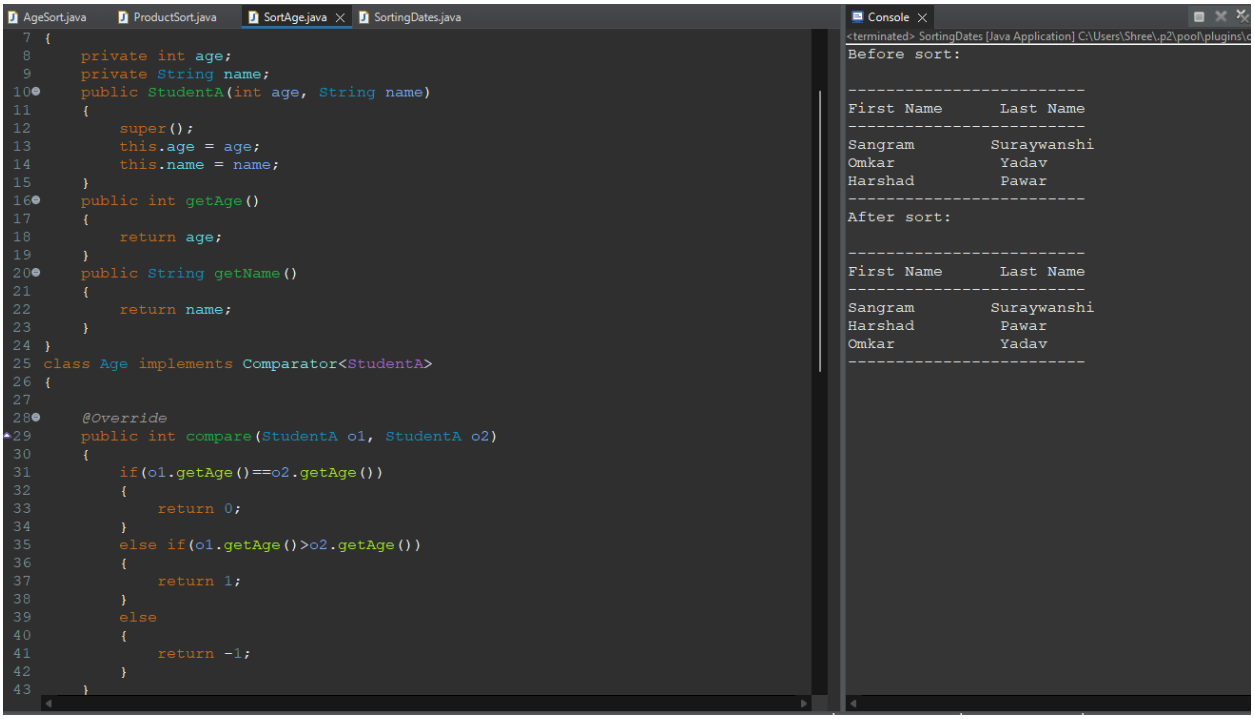
```
AgeSort.java   ProductSort.java   SortAge.java

 4 import java.util.Collections;
 5 import java.util.Comparator;
 6 class StudentA
 7 {
 8     private int age;
 9     private String name;
10     public StudentA(int age, String name)
11     {
12         super();
13         this.age = age;
14         this.name = name;
15     }
16     public int getAge()
17     {
18         return age;
19     }
20     public String getName()
21     {
22         return name;
23     }
24 }
25 class Age implements Comparator<StudentA>
26 {
27
28     @Override
29     public int compare(StudentA o1, StudentA o2)
30     {
31         if(o1.getAge()==o2.getAge())
32         {
33             return 0;
34         }
35         else if(o1.getAge()>o2.getAge())
36         {
37             return 1;
38         }
39         else
40         {
```

```
Console
<terminated> SortAge [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.j
Before sort:


--------------------------------------------
Student Age       Student Name
--------------------------------------------
18                sangram
23                Harshad
20                omkar
25                aditya
--------------------------------------------

After sort:


--------------------------------------------
Student Age       Student Name
--------------------------------------------
18                sangram
20                omkar
23                Harshad
25                aditya
--------------------------------------------
```

```
AgeSort.java   ProductSort.java   SortAge.java

43     }
44
45 }
46 public class SortAge
47 {
48
49     public static void main(String[] args)
50     {
51         ArrayList<StudentA> a1 = new ArrayList<StudentA>();
52         a1.add(new StudentA(18,"sangram"));
53         a1.add(new StudentA(23,"Harshad"));
54         a1.add(new StudentA(20,"omkar"));
55         a1.add(new StudentA(25,"aditya"));
56         System.out.println("Before sort: ");
57         System.out.println("");
58         System.out.println("");
59         System.out.println("--------------------------------------------");
60         System.out.println("Student Age\t Student Name");
61         System.out.println("--------------------------------------------");
62         for(StudentA e:a1)
63         {
64             System.out.println(e.getAge()+"\t\t"+e.getName());
65         }
66         System.out.println("--------------------------------------------");
67         Collections.sort(a1, new Age());
68         System.out.println("");
69         System.out.println("After sort: ");
70         System.out.println("");
71         System.out.println("");
72         System.out.println("--------------------------------------------");
73         System.out.println("Student Age\t Student Name");
74         System.out.println("--------------------------------------------");
75         for(StudentA e:a1)
76         {
77             System.out.println(e.getAge()+"\t\t"+e.getName());
78         }
79         System.out.println("--------------------------------------------");
```

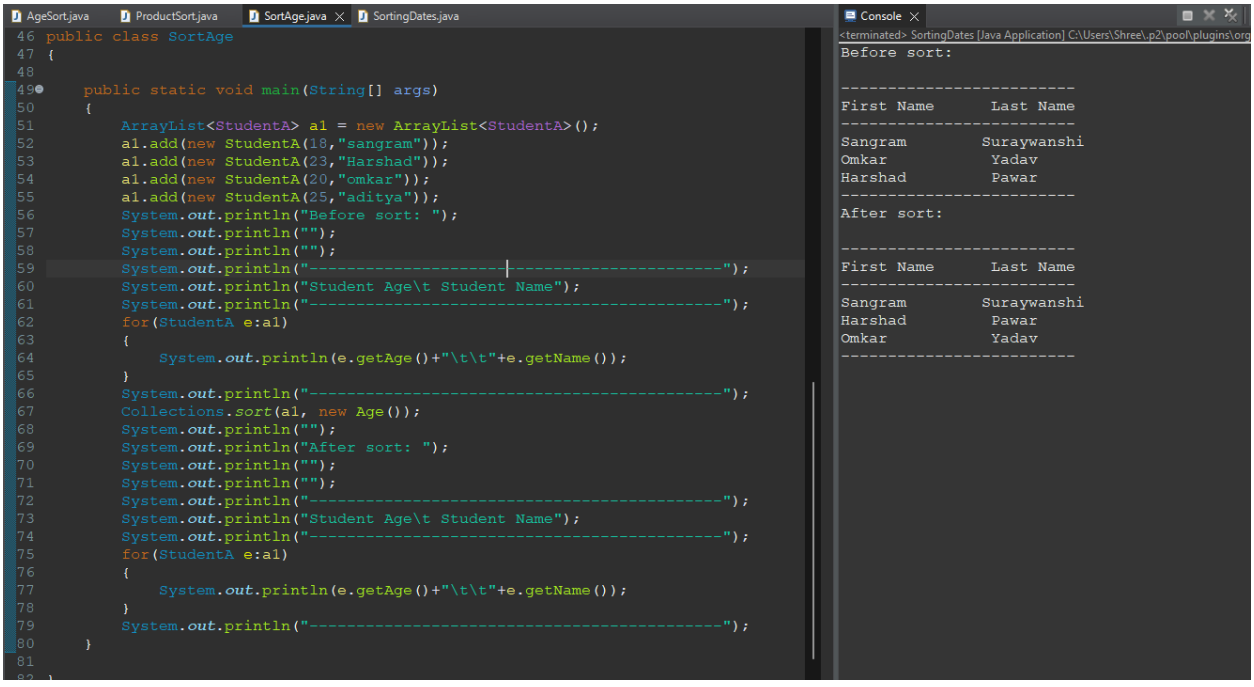
```
Console
<terminated> SortAge [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.j
Before sort:


--------------------------------------------
Student Age       Student Name
--------------------------------------------
18                sangram
23                Harshad
20                omkar
25                aditya
--------------------------------------------

After sort:


--------------------------------------------
Student Age       Student Name
--------------------------------------------
18                sangram
20                omkar
23                Harshad
25                aditya
--------------------------------------------
```

Q.15 Create a Person class and implement a Comparator to sort instances by their last names.



```
AgeSort.java  ProductSort.java  SortAge.java  SortingDates.java

 7  {
 8      private int age;
 9      private String name;
10      public StudentA(int age, String name)
11      {
12          super();
13          this.age = age;
14          this.name = name;
15      }
16      public int getAge()
17      {
18          return age;
19      }
20      public String getName()
21      {
22          return name;
23      }
24  }
25  class Age implements Comparator<StudentA>
26  {
27
28      @Override
29      public int compare(StudentA o1, StudentA o2)
30      {
31          if(o1.getAge()==o2.getAge())
32          {
33              return 0;
34          }
35          else if(o1.getAge()>o2.getAge())
36          {
37              return 1;
38          }
39          else
40          {
41              return -1;
42          }
43      }
```

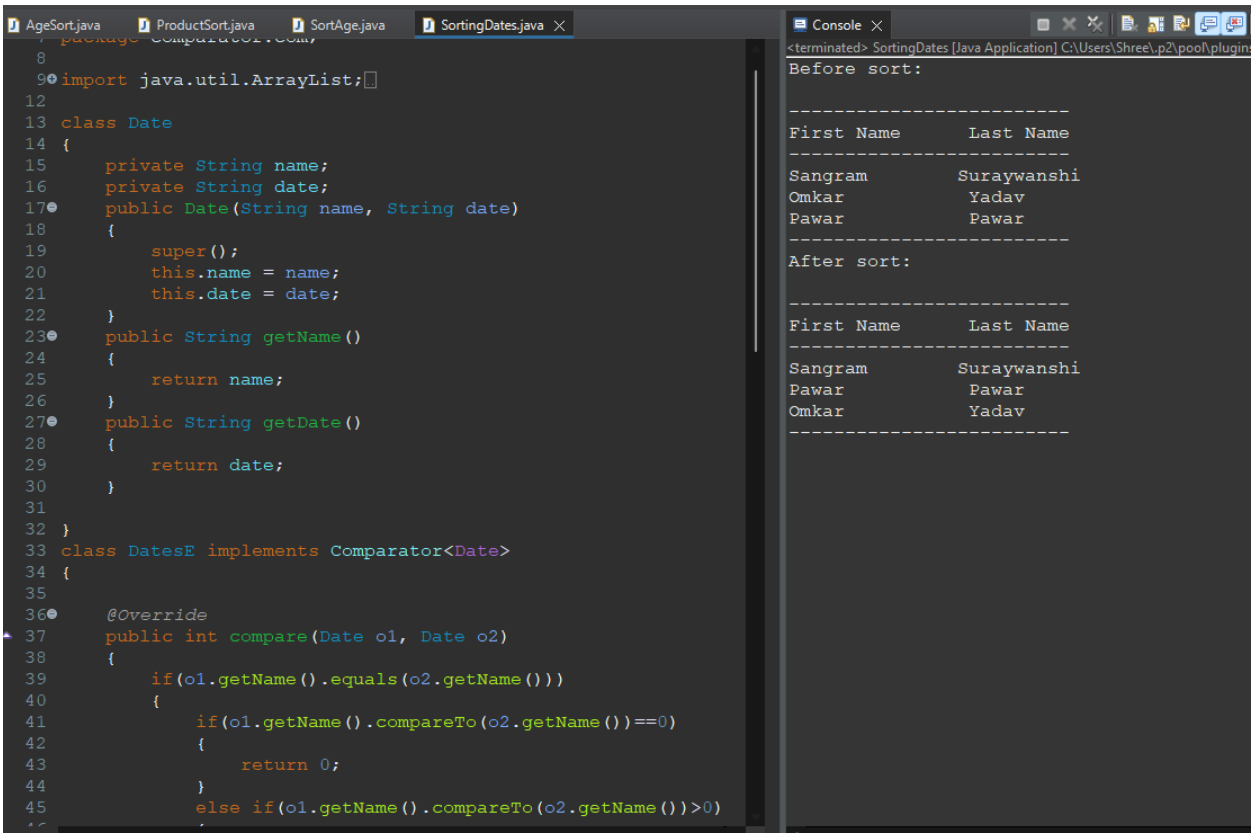
```
Console
<terminated> SortingDates [Java Application] C:\Users\Shree\.p2\pool\plugins\c
Before sort:

-------------------------
First Name      Last Name
-------------------------
Sangram        Suraywanshi
Omkar           Yadav
Harshad         Pawar
-------------------------
After sort:

-------------------------
First Name      Last Name
-------------------------
Sangram        Suraywanshi
Harshad         Pawar
Omkar           Yadav
-------------------------
```



```
AgeSort.java  ProductSort.java  SortAge.java  SortingDates.java

46  public class SortAge
47  {
48
49      public static void main(String[] args)
50      {
51          ArrayList<StudentA> a1 = new ArrayList<StudentA>();
52          a1.add(new StudentA(18,"sangram"));
53          a1.add(new StudentA(23,"Harshad"));
54          a1.add(new StudentA(20,"omkar"));
55          a1.add(new StudentA(25,"aditya"));
56          System.out.println("Before sort: ");
57          System.out.println("");
58          System.out.println("");
59          System.out.println("----------------------------------------");
60          System.out.println("Student Age\t Student Name");
61          System.out.println("----------------------------------------");
62          for(StudentA e:a1)
63          {
64              System.out.println(e.getAge()+"\t\t"+e.getName());
65          }
66          System.out.println("----------------------------------------");
67          Collections.sort(a1, new Age());
68          System.out.println("");
69          System.out.println("After sort: ");
70          System.out.println("");
71          System.out.println("");
72          System.out.println("----------------------------------------");
73          System.out.println("Student Age\t Student Name");
74          System.out.println("----------------------------------------");
75          for(StudentA e:a1)
76          {
77              System.out.println(e.getAge()+"\t\t"+e.getName());
78          }
79          System.out.println("----------------------------------------");
80      }
81
```

```
Console
<terminated> SortingDates [Java Application] C:\Users\Shree\.p2\pool\plugins\org
Before sort:

-------------------------
First Name      Last Name
-------------------------
Sangram        Suraywanshi
Omkar           Yadav
Harshad         Pawar
-------------------------
After sort:

-------------------------
First Name      Last Name
-------------------------
Sangram        Suraywanshi
Harshad         Pawar
Omkar           Yadav
-------------------------
```

Q.16 Write a Comparator to sort Student objects by their last name, and if last names are the same, sort by first name.



```
AgeSort.java | ProductSort.java | SortAge.java | SortingDates.java

 8
 9 import java.util.ArrayList;
12
13 class Date
14 {
15     private String name;
16     private String date;
17     public Date(String name, String date)
18     {
19         super();
20         this.name = name;
21         this.date = date;
22     }
23     public String getName()
24     {
25         return name;
26     }
27     public String getDate()
28     {
29         return date;
30     }
31
32 }
33 class DatesE implements Comparator<Date>
34 {
35
36     @Override
37     public int compare(Date o1, Date o2)
38     {
39         if(o1.getName().equals(o2.getName()))
40         {
41             if(o1.getName().compareTo(o2.getName())==0)
42             {
43                 return 0;
44             }
45             else if(o1.getName().compareTo(o2.getName())>0)
```

```
Console
<terminated> SortingDates [Java Application] C:\Users\Shree\.p2\pool\plugins
Before sort:

-------------------------
First Name      Last Name
-------------------------
Sangram        Suraywanshi
Omkar           Yadav
Pawar           Pawar
-------------------------
After sort:

-------------------------
First Name      Last Name
-------------------------
Sangram        Suraywanshi
Pawar           Pawar
Omkar           Yadav
-------------------------
```

AgeSort.java | ProductSort.java | SortAge.java | SortingDates.java

```
            }
            else if(o1.getName().compareTo(o2.getName())>0)
            {
                return 1;
            }
            else
            {
                return -1;
            }
        }
        else
        {
            if(o1.getDate().compareTo(o2.getDate())==0)
            {
                return 0;
            }
            else if(o1.getDate().compareTo(o2.getDate())>0)
            {
                return 1;
            }
            else
            {
                return -1;
            }
        }
    }

}
public class SortingDates
{

    public static void main(String[] args)
    {
        ArrayList<Date> a1 = new ArrayList<Date>();
        a1.add(new Date("Sangram","       Suraywanshi"));
        a1.add(new Date("Omkar    ","Yadav"));
        a1.add(new Date("Pawar    ","Pawar"));
```

Console

```
<terminated> SortingDates [Java Application] C:\Users\Shree\.p2\pool\plugins\or
Before sort:

-------------------------
First Name       Last Name
-------------------------
Sangram         Suraywanshi
Omkar            Yadav
Pawar            Pawar
-------------------------
After sort:

-------------------------
First Name       Last Name
-------------------------
Sangram         Suraywanshi
Pawar            Pawar
Omkar            Yadav
-------------------------
```

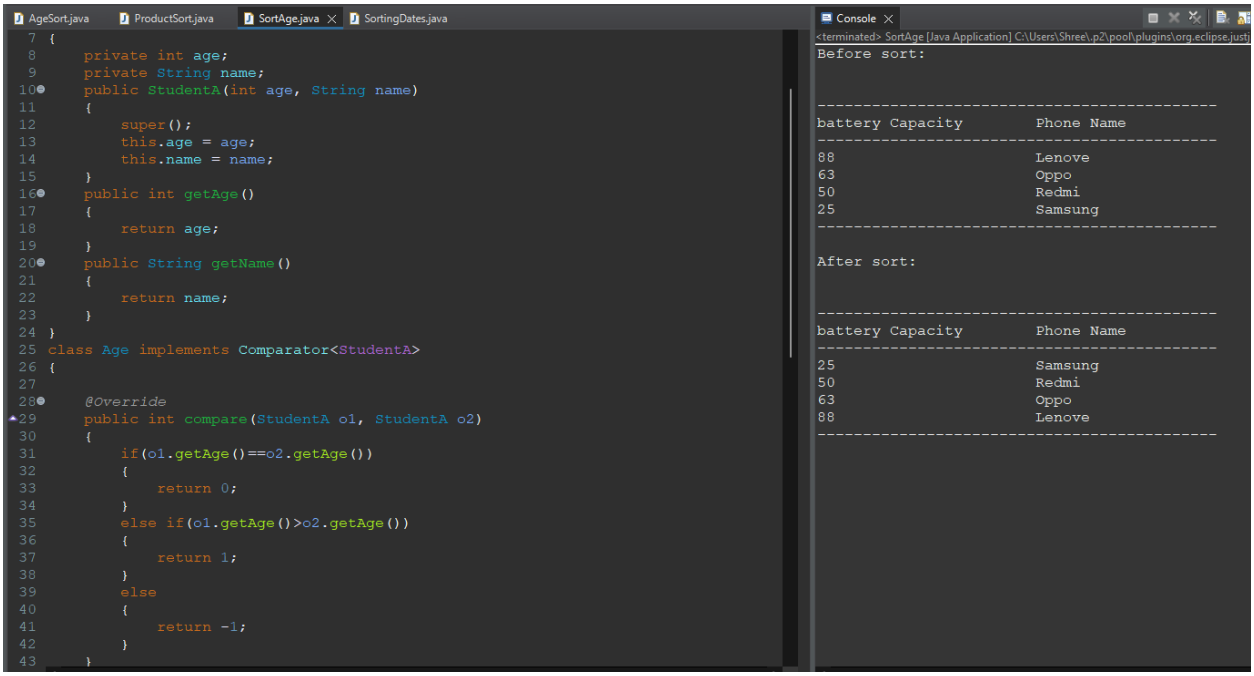
AgeSort.java | ProductSort.java | SortAge.java | SortingDates.java

```
    }

}
public class SortingDates
{

    public static void main(String[] args)
    {
        ArrayList<Date> a1 = new ArrayList<Date>();
        a1.add(new Date("Sangram","       Suraywanshi"));
        a1.add(new Date("Omkar    ","Yadav"));
        a1.add(new Date("Pawar    ","Pawar"));
        System.out.println("Before sort: ");
        System.out.println("");
        System.out.println("-------------------------");
        System.out.println("First Name \tLast Name");
        System.out.println("-------------------------");
        for(Date e:a1)
        {
            System.out.println(e.getName()+"\t"+e.getDate());
        }
        System.out.println("-------------------------");
        Collections.sort(a1, new DatesE());
        System.out.println("After sort: ");
        System.out.println("");
        System.out.println("-------------------------");
        System.out.println("First Name \tLast Name");
        System.out.println("-------------------------");
        for(Date e:a1)
        {
            System.out.println(e.getName()+"\t"+e.getDate());
        }
        System.out.println("-------------------------");
    }

}
```

Console

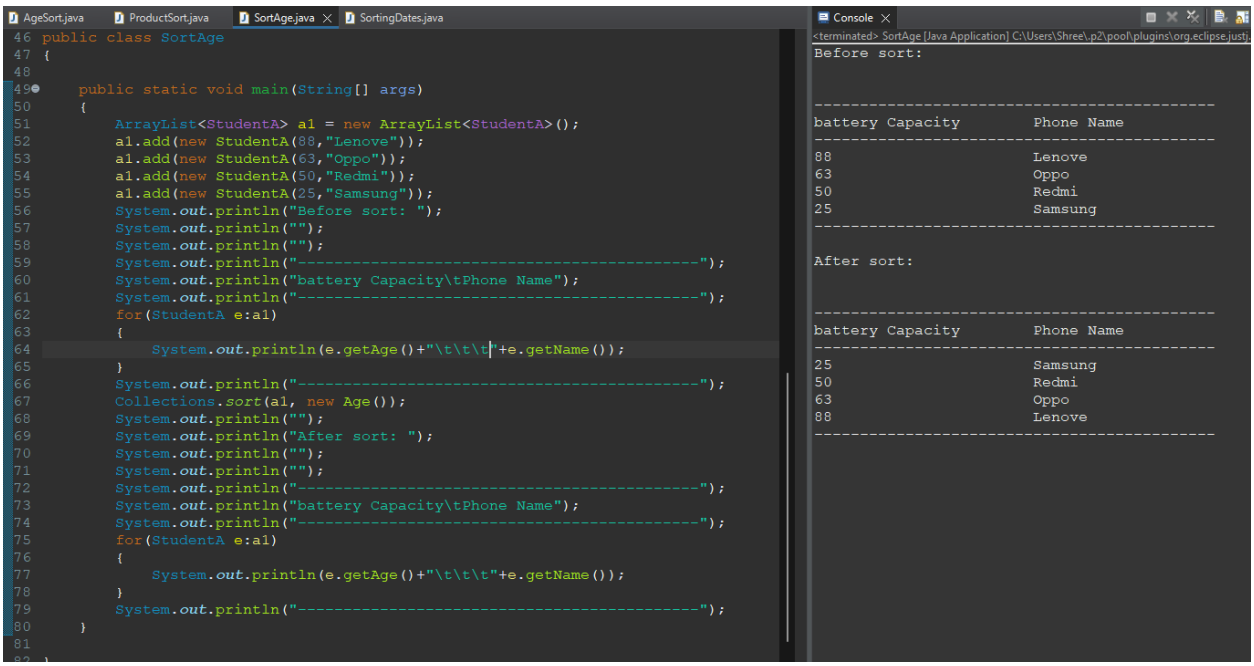
```
<terminated> SortingDates [Java Application] C:\Users\Shree\.p2\pool\plugi
Before sort:

-------------------------
First Name       Last Name
-------------------------
Sangram         Suraywanshi
Omkar            Yadav
Pawar            Pawar
-------------------------
After sort:

-------------------------
First Name       Last Name
-------------------------
Sangram         Suraywanshi
Pawar            Pawar
Omkar            Yadav
-------------------------
```

Q.17 Create a Phone class and implement a Comparator to sort phones by battery capacity.



```
AgeSort.java  ProductSort.java  SortAge.java  SortingDates.java
 7 {
 8     private int age;
 9     private String name;
10     public StudentA(int age, String name)
11     {
12         super();
13         this.age = age;
14         this.name = name;
15     }
16     public int getAge()
17     {
18         return age;
19     }
20     public String getName()
21     {
22         return name;
23     }
24 }
25 class Age implements Comparator<StudentA>
26 {
27
28     @Override
29     public int compare(StudentA o1, StudentA o2)
30     {
31         if(o1.getAge()==o2.getAge())
32         {
33             return 0;
34         }
35         else if(o1.getAge()>o2.getAge())
36         {
37             return 1;
38         }
39         else
40         {
41             return -1;
42         }
43     }
```

```
Console
<terminated> SortAge [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj
Before sort:


-------------------------------------------
battery Capacity        Phone Name
-------------------------------------------
88                      Lenove
63                      Oppo
50                      Redmi
25                      Samsung
-------------------------------------------

After sort:


-------------------------------------------
battery Capacity        Phone Name
-------------------------------------------
25                      Samsung
50                      Redmi
63                      Oppo
88                      Lenove
-------------------------------------------
```



```
AgeSort.java  ProductSort.java  SortAge.java  SortingDates.java
46 public class SortAge
47 {
48
49     public static void main(String[] args)
50     {
51         ArrayList<StudentA> a1 = new ArrayList<StudentA>();
52         a1.add(new StudentA(88,"Lenove"));
53         a1.add(new StudentA(63,"Oppo"));
54         a1.add(new StudentA(50,"Redmi"));
55         a1.add(new StudentA(25,"Samsung"));
56         System.out.println("Before sort: ");
57         System.out.println("");
58         System.out.println("");
59         System.out.println("-------------------------------------------");
60         System.out.println("battery Capacity\tPhone Name");
61         System.out.println("-------------------------------------------");
62         for(StudentA e:a1)
63         {
64             System.out.println(e.getAge()+"\t\t\t"+e.getName());
65         }
66         System.out.println("-------------------------------------------");
67         Collections.sort(a1, new Age());
68         System.out.println("");
69         System.out.println("After sort: ");
70         System.out.println("");
71         System.out.println("");
72         System.out.println("-------------------------------------------");
73         System.out.println("battery Capacity\tPhone Name");
74         System.out.println("-------------------------------------------");
75         for(StudentA e:a1)
76         {
77             System.out.println(e.getAge()+"\t\t\t"+e.getName());
78         }
79         System.out.println("-------------------------------------------");
80     }
```

```
Console
<terminated> SortAge [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclipse.justj
Before sort:


-------------------------------------------
battery Capacity        Phone Name
-------------------------------------------
88                      Lenove
63                      Oppo
50                      Redmi
25                      Samsung
-------------------------------------------

After sort:


-------------------------------------------
battery Capacity        Phone Name
-------------------------------------------
25                      Samsung
50                      Redmi
63                      Oppo
88                      Lenove
-------------------------------------------
```

Q.18 Implement a Comparator to sort a list of Order objects by order date.

AgeSort.java | ProductSort.java | SortAge.java | SortEvenDuration.java | SortingDates.java

```
import java.util.ArrayList;

class Date
{
    private String name;
    private String date;
    public Date(String name, String date)
    {
        super();
        this.name = name;
        this.date = date;
    }
    public String getName()
    {
        return name;
    }
    public String getDate()
    {
        return date;
    }

}
class DatesE implements Comparator<Date>
{

    @Override
    public int compare(Date o1, Date o2)
    {
        if(o1.getDate().compareTo(o2.getDate())==0)
        {
            return 0;
        }
        else if(o1.getDate().compareTo(o2.getDate())>0)
        {
            return 1;
        }
        else
```

Console

```
<terminated> SortingDates [Java Application] C:\Users\Shre
Before sort:

-------------------------
Event Name      Event Date
-------------------------
Reception       1/1/2024
Mendi           2/4/2024
Haldi           1/3/2024
-------------------------
After sort:

-------------------------
Event Name      Event Date
-------------------------
Reception       1/1/2024
Haldi           1/3/2024
Mendi           2/4/2024
-------------------------
```

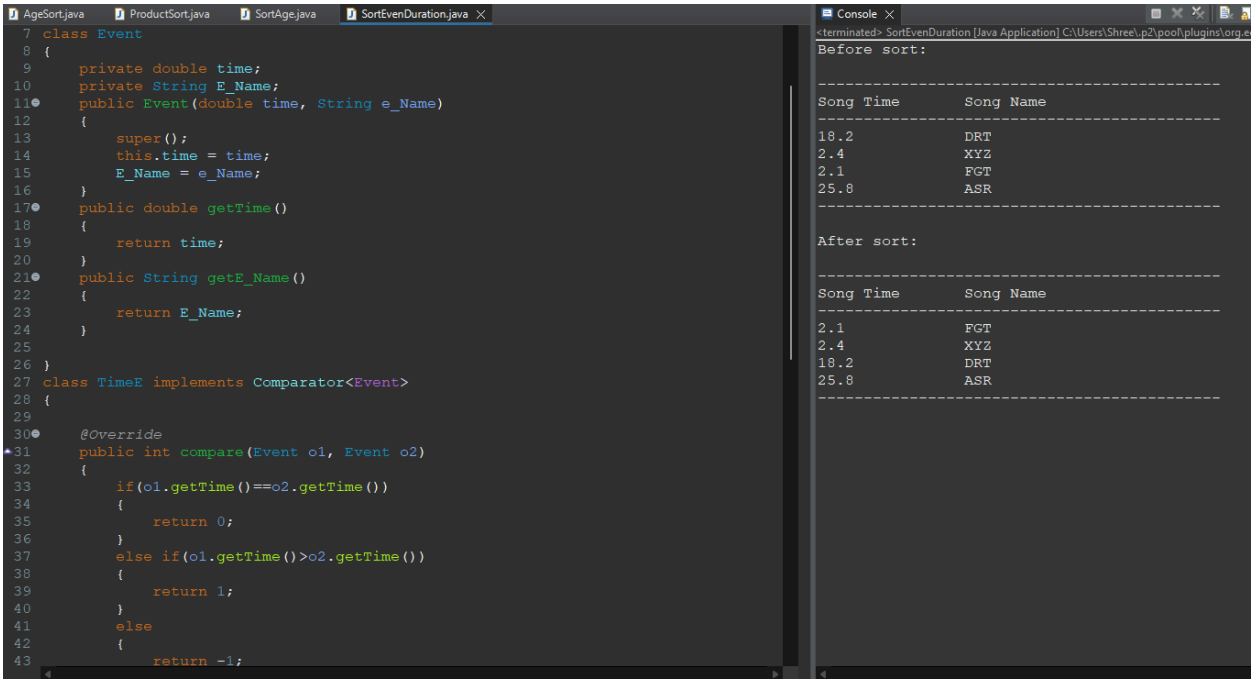
AgeSort.java | ProductSort.java | SortAge.java | SortEvenDuration.java | SortingDates.java

```
        else
        {
            return -1;
        }
    }

}
public class SortingDates
{

    public static void main(String[] args)
    {
        ArrayList<Date> a1 = new ArrayList<Date>();
        a1.add(new Date("Reception","1/1/2024"));
        a1.add(new Date("Mendi    ","2/4/2024"));
        a1.add(new Date("Haldi    ","1/3/2024"));
        System.out.println("Before sort: ");
        System.out.println("");
        System.out.println("-------------------------");
        System.out.println("Event Name \tEvent Date");
        System.out.println("-------------------------");
        for(Date e:a1)
        {
            System.out.println(e.getName()+"\t"+e.getDate());
        }
        System.out.println("-------------------------");
        Collections.sort(a1, new DatesE());
        System.out.println("After sort: ");
        System.out.println("");
        System.out.println("-------------------------");
        System.out.println("Event Name \tEvent Date");
        System.out.println("-------------------------");
        for(Date e:a1)
        {
            System.out.println(e.getName()+"\t"+e.getDate());
        }
        System.out.println("-------------------------");
```

Console

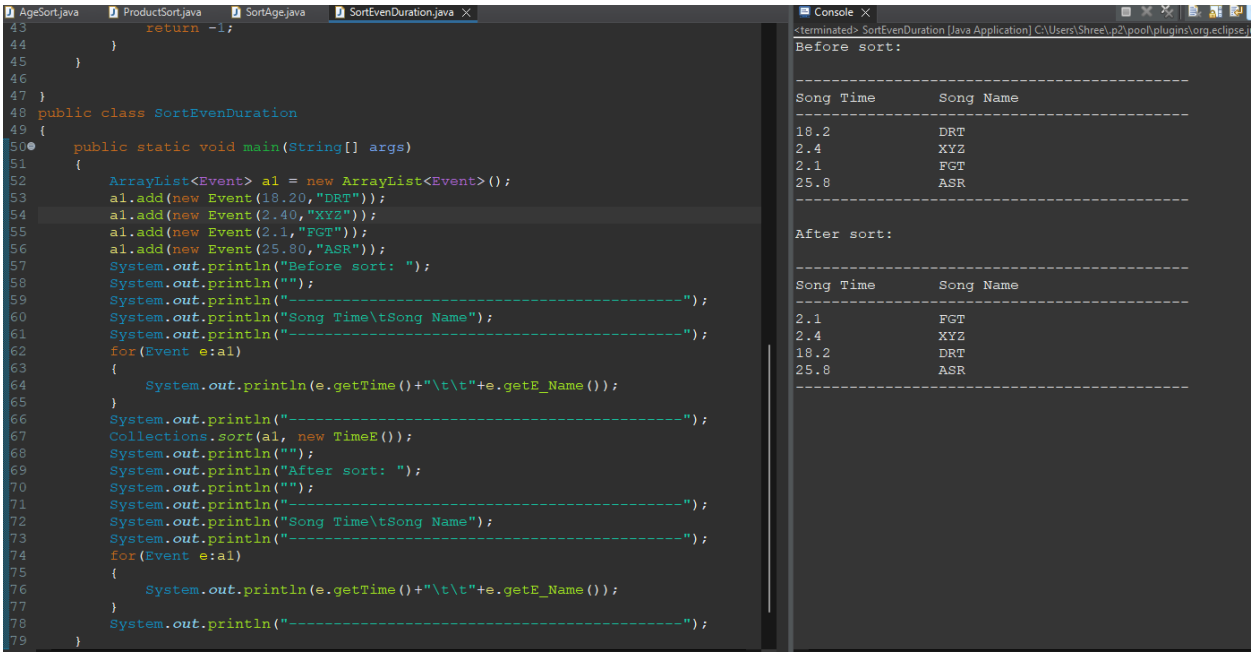
```
<terminated> SortingDates [Java Application] C:\Users\Shre
Before sort:

-------------------------
Event Name      Event Date
-------------------------
Reception       1/1/2024
Mendi           2/4/2024
Haldi           1/3/2024
-------------------------
After sort:

-------------------------
Event Name      Event Date
-------------------------
Reception       1/1/2024
Haldi           1/3/2024
Mendi           2/4/2024
-------------------------
```

Q.19 Write a Comparator to sort Song objects by artist name in ascending order.



```
class Event
{
    private double time;
    private String E_Name;
    public Event(double time, String e_Name)
    {
        super();
        this.time = time;
        E_Name = e_Name;
    }
    public double getTime()
    {
        return time;
    }
    public String getE_Name()
    {
        return E_Name;
    }

}
class TimeE implements Comparator<Event>
{

    @Override
    public int compare(Event o1, Event o2)
    {
        if(o1.getTime()==o2.getTime())
        {
            return 0;
        }
        else if(o1.getTime()>o2.getTime())
        {
            return 1;
        }
        else
        {
            return -1;
```

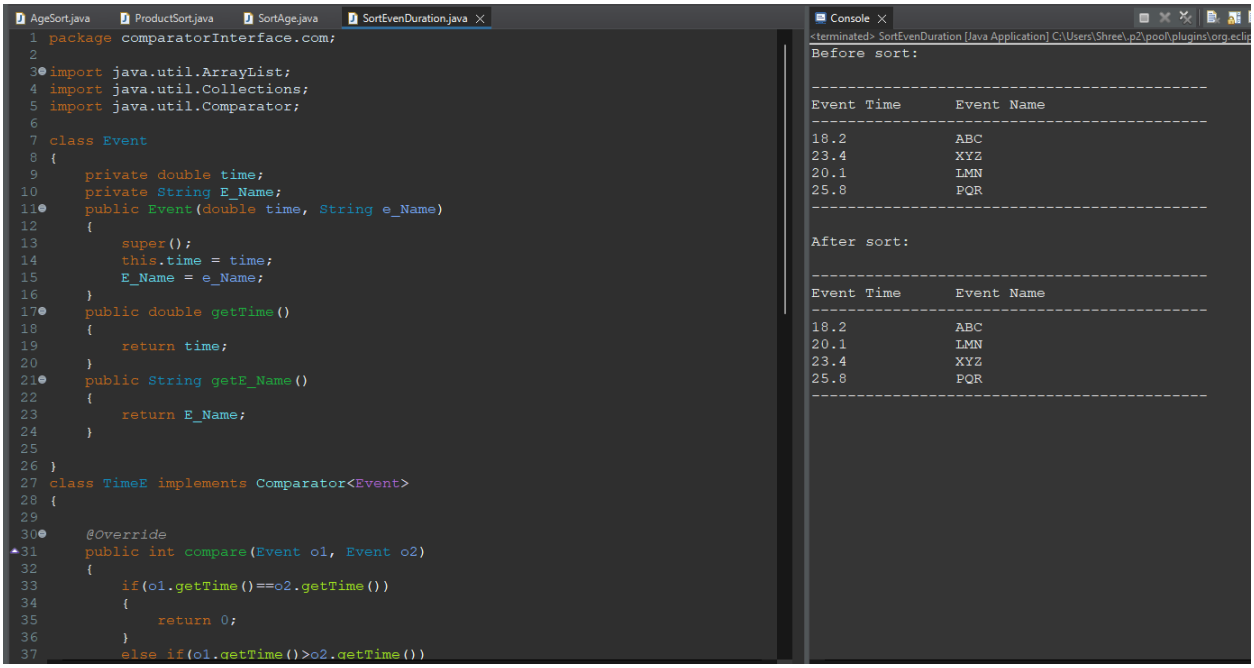
```
Before sort:

-------------------------------------------
Song Time       Song Name
-------------------------------------------
18.2            DRT
2.4             XYZ
2.1             FGT
25.8            ASR
-------------------------------------------

After sort:

-------------------------------------------
Song Time       Song Name
-------------------------------------------
2.1             FGT
2.4             XYZ
18.2            DRT
25.8            ASR
-------------------------------------------
```



```
            return -1;
        }
    }

}
public class SortEvenDuration
{
    public static void main(String[] args)
    {
        ArrayList<Event> a1 = new ArrayList<Event>();
        a1.add(new Event(18.20,"DRT"));
        a1.add(new Event(2.40,"XYZ"));
        a1.add(new Event(2.1,"FGT"));
        a1.add(new Event(25.80,"ASR"));
        System.out.println("Before sort: ");
        System.out.println("");
        System.out.println("-------------------------------------------");
        System.out.println("Song Time\tSong Name");
        System.out.println("-------------------------------------------");
        for(Event e:a1)
        {
            System.out.println(e.getTime()+"\t\t"+e.getE_Name());
        }
        System.out.println("-------------------------------------------");
        Collections.sort(a1, new TimeE());
        System.out.println("");
        System.out.println("After sort: ");
        System.out.println("");
        System.out.println("-------------------------------------------");
        System.out.println("Song Time\tSong Name");
        System.out.println("-------------------------------------------");
        for(Event e:a1)
        {
            System.out.println(e.getTime()+"\t\t"+e.getE_Name());
        }
        System.out.println("-------------------------------------------");
    }
```

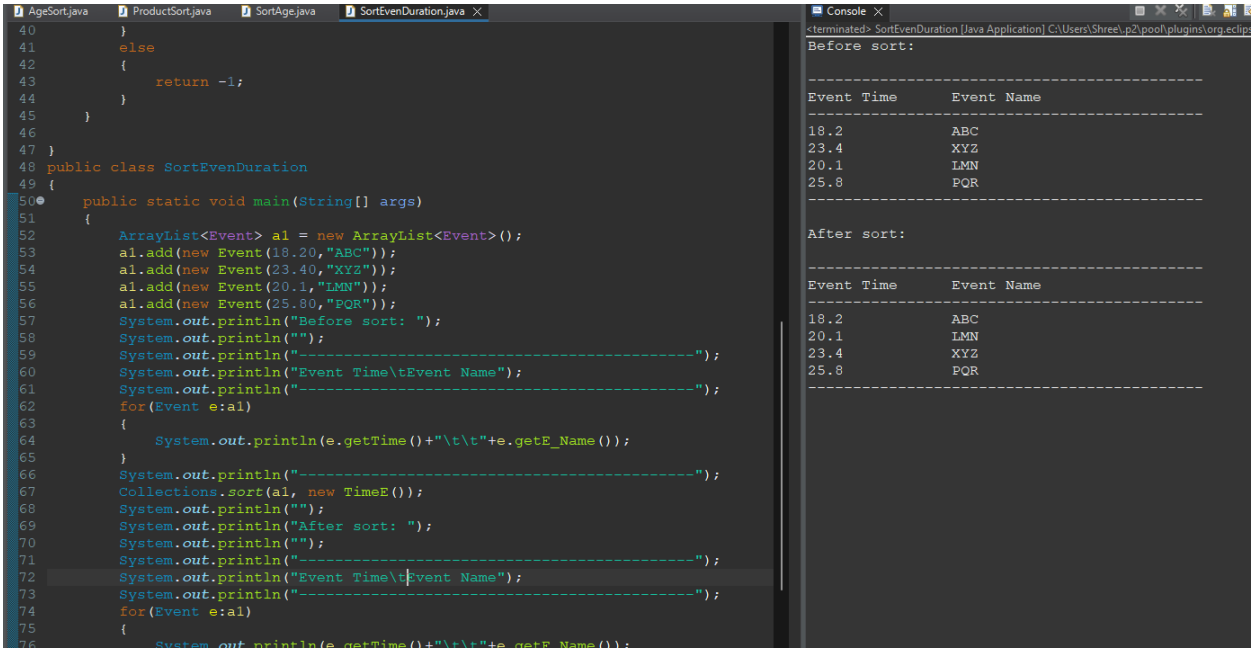
```
Before sort:

-------------------------------------------
Song Time       Song Name
-------------------------------------------
18.2            DRT
2.4             XYZ
2.1             FGT
25.8            ASR
-------------------------------------------

After sort:

-------------------------------------------
Song Time       Song Name
-------------------------------------------
2.1             FGT
2.4             XYZ
18.2            DRT
25.8            ASR
-------------------------------------------
```

Q.20 Create a Event class and implement a Comparator to sort events by duration



```
AgeSort.java  ProductSort.java  SortAge.java  SortEvenDuration.java

1  package comparatorInterface.com;
2
3  import java.util.ArrayList;
4  import java.util.Collections;
5  import java.util.Comparator;
6
7  class Event
8  {
9      private double time;
10     private String E_Name;
11     public Event(double time, String e_Name)
12     {
13         super();
14         this.time = time;
15         E_Name = e_Name;
16     }
17     public double getTime()
18     {
19         return time;
20     }
21     public String getE_Name()
22     {
23         return E_Name;
24     }
25
26 }
27 class TimeE implements Comparator<Event>
28 {
29
30     @Override
31     public int compare(Event o1, Event o2)
32     {
33         if(o1.getTime()==o2.getTime())
34         {
35             return 0;
36         }
37         else if(o1.getTime()>o2.getTime())
```

```
Console
<terminated> SortEvenDuration [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclip
Before sort:

-------------------------------------------
Event Time      Event Name
-------------------------------------------
18.2            ABC
23.4            XYZ
20.1            LMN
25.8            PQR
-------------------------------------------

After sort:

-------------------------------------------
Event Time      Event Name
-------------------------------------------
18.2            ABC
20.1            LMN
23.4            XYZ
25.8            PQR
-------------------------------------------
```



```
AgeSort.java  ProductSort.java  SortAge.java  SortEvenDuration.java

40         }
41         else
42         {
43             return -1;
44         }
45     }
46
47 }
48 public class SortEvenDuration
49 {
50     public static void main(String[] args)
51     {
52         ArrayList<Event> a1 = new ArrayList<Event>();
53         a1.add(new Event(18.20,"ABC"));
54         a1.add(new Event(23.40,"XYZ"));
55         a1.add(new Event(20.1,"LMN"));
56         a1.add(new Event(25.80,"PQR"));
57         System.out.println("Before sort: ");
58         System.out.println("");
59         System.out.println("-------------------------------------------");
60         System.out.println("Event Time\tEvent Name");
61         System.out.println("-------------------------------------------");
62         for(Event e:a1)
63         {
64             System.out.println(e.getTime()+"\t\t"+e.getE_Name());
65         }
66         System.out.println("-------------------------------------------");
67         Collections.sort(a1, new TimeE());
68         System.out.println("");
69         System.out.println("After sort: ");
70         System.out.println("");
71         System.out.println("-------------------------------------------");
72         System.out.println("Event Time\tEvent Name");
73         System.out.println("-------------------------------------------");
74         for(Event e:a1)
75         {
76             System.out.println(e.getTime()+"\t\t"+e.getE_Name());
```

```
Console
<terminated> SortEvenDuration [Java Application] C:\Users\Shree\.p2\pool\plugins\org.eclips
Before sort:

-------------------------------------------
Event Time      Event Name
-------------------------------------------
18.2            ABC
23.4            XYZ
20.1            LMN
25.8            PQR
-------------------------------------------

After sort:

-------------------------------------------
Event Time      Event Name
-------------------------------------------
18.2            ABC
20.1            LMN
23.4            XYZ
25.8            PQR
-------------------------------------------
```