

Experiment No. 7

Aim_: Create a class Product having private data members product_name, cost, manufacturer, max_discount. Include appropriate methods to set the member values and override toString method to display the data members. Create an arraylist to store 10 product objects. Take the input from file using BufferedReader. Write a program to

- 1) Display the list of products using iterator
- 2) Display the list of products whose max_discount is 50%. Also display the final cost at which the product can be given.
- 3) Products sorted according to the cost
- 4) Products sorted according to the manufacturer

Code :

Product.java

```
class Product {
    private String pname, manuf;
    private double cost;
    private double maxDis;
    Product(String pname, double cost, String manuf, double maxDis){
        this.pname = pname;
        this.cost = cost;
        this.manuf = manuf;
        this.maxDis = maxDis;
    }
    double getMaxDis(){
        return maxDis;
    }
    double getCost(){
        return cost;
    }
    String getManf(){
        return manuf;
    }
    public String toString(){
        return ("\nproduct name : " + pname + " || cost : " + cost + " ||
Manufacturer: " + manuf + " || Max Discount : " + maxDis + "%"+ "\n");
    }
}
```

ManfSorter.java

```
import java.util.Comparator;

public class ManfSorter implements Comparator<Product> {
    public int compare(Product p1, Product p2) {
        return (p1.getManf().compareTo(p2.getManf()));
    }
}
```

Date :- 07-07-2023

CostSorter.java

```
import java.util.Comparator;

public class CostSorter implements Comparator<Product> {
    public int compare(Product p1, Product p2) {
        return (p1.getCost() < p2.getCost() ? -1
                : (p1.getCost() > p2.getCost() ? 1
                    : p1.getMaxDis() < p2.getMaxDis() ? -1
                      : p1.getMaxDis() > p2.getMaxDis() ? 1 :
p1.getPro().compareTo(p2.getPro())));
    }
}
```

Test.java

```
import java.util.*;
import java.io.*;

class Test{
    public static void main(String args[]){
        ArrayList<Product> prList = new ArrayList<Product>();

        try(BufferedReader br = new BufferedReader(new
FileReader("Product.txt"))){
            String str[], line;
            while((line = br.readLine()) != null){
                str = line.split(",");
                prList.add(new Product(str[0], Double.parseDouble(str[1]),
str[2], Double.parseDouble(str[3])));
            }
        }
        catch(Exception e){
            e.printStackTrace();
        }

        // System.out.println(prList);
        // Iterator<Product> itr = prList.iterator();

        // while(itr.hasNext()){
        //     System.out.println(itr.next() + " ");
        // }

        // System.out.println("Product which has max Discount 50% are below :
");

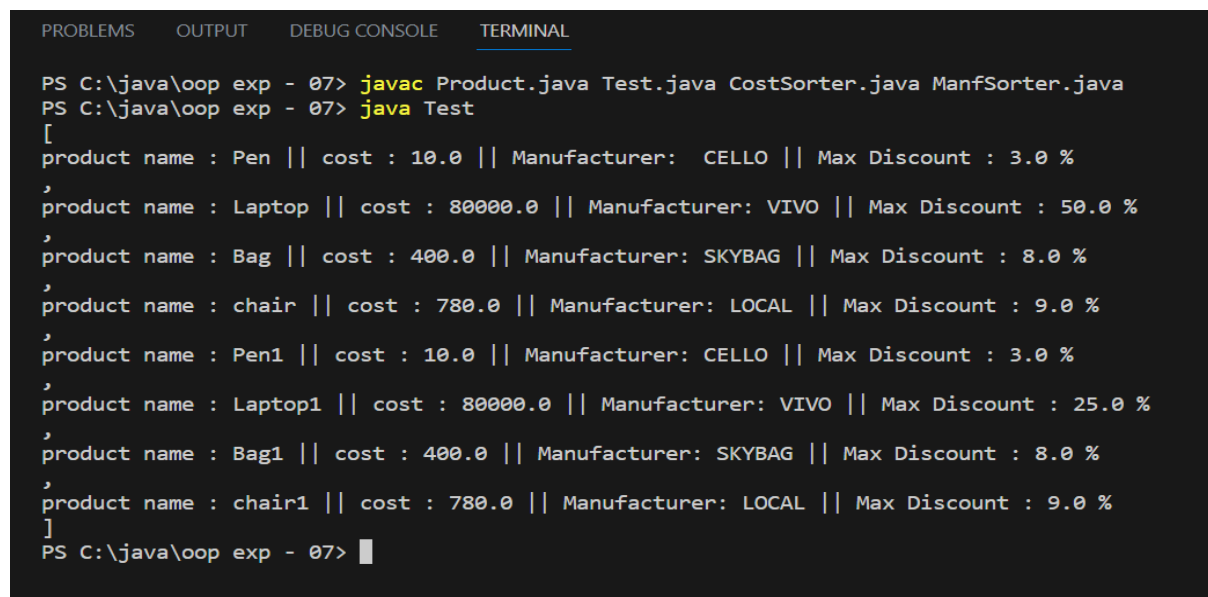
        // for(int i = 0; i < prList.size(); i++){
        //     if(prList.get(i).getMaxDis() == 50.0){
```

Date :- 07-07-2023

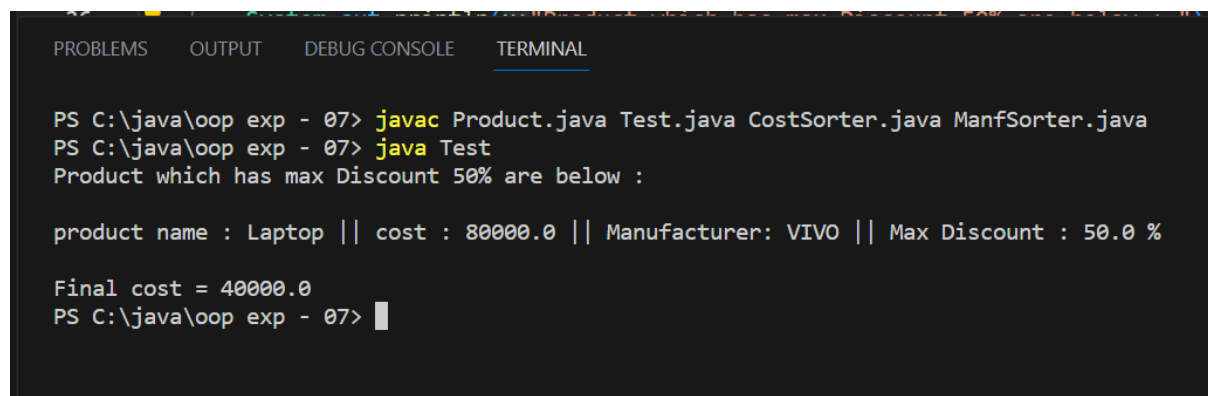
```
//      System.out.println(prList.get(i));
//      System.out.println("Final cost = " +
(0.5*prList.get(i).getCost()));
//  }
//  }
System.out.println("Sorting Based on Cost if same then on the basis of
max discount and if also same then on the basis of product name : ");
Collections.sort(prList, new CostSorter());
// System.out.println("Sorting Based on manufacturer : ");
// Collections.sort(prList, new ManfSorter());
System.out.println(prList);
}
}
```

Test Cases(ScreenShot Output) :

1. Input : File containing details such as product name, cost, manufacturer, and max_discount
e.g. Pen,50,Parker,5 Laptop,65000,HP,50 Bag,500,Skybags,40



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\java\oop exp - 07> javac Product.java Test.java CostSorter.java ManfSorter.java
PS C:\java\oop exp - 07> java Test
[
product name : Pen || cost : 10.0 || Manufacturer: CELLO || Max Discount : 3.0 %
,
product name : Laptop || cost : 80000.0 || Manufacturer: VIVO || Max Discount : 50.0 %
,
product name : Bag || cost : 400.0 || Manufacturer: SKYBAG || Max Discount : 8.0 %
,
product name : chair || cost : 780.0 || Manufacturer: LOCAL || Max Discount : 9.0 %
,
product name : Pen1 || cost : 10.0 || Manufacturer: CELLO || Max Discount : 3.0 %
,
product name : Laptop1 || cost : 80000.0 || Manufacturer: VIVO || Max Discount : 25.0 %
,
product name : Bag1 || cost : 400.0 || Manufacturer: SKYBAG || Max Discount : 8.0 %
,
product name : chair1 || cost : 780.0 || Manufacturer: LOCAL || Max Discount : 9.0 %
]
PS C:\java\oop exp - 07> █
```



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
PS C:\java\oop exp - 07> javac Product.java Test.java CostSorter.java ManfSorter.java
PS C:\java\oop exp - 07> java Test
Product which has max Discount 50% are below :

product name : Laptop || cost : 80000.0 || Manufacturer: VIVO || Max Discount : 50.0 %

Final cost = 40000.0
PS C:\java\oop exp - 07> █
```

Date :- 07-07-2023

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\java\oop exp - 07> javac Product.java Test.java CostSorter.java ManfSorter.java
PS C:\java\oop exp - 07> java Test
Sorting Based on Cost :
[
product name : Pen || cost : 10.0 || Manufacturer: CELLO || Max Discount : 3.0 %
,
product name : Pen1 || cost : 10.0 || Manufacturer: CELLO || Max Discount : 3.0 %
,
product name : Bag || cost : 400.0 || Manufacturer: SKYBAG || Max Discount : 8.0 %
,
product name : Bag1 || cost : 400.0 || Manufacturer: SKYBAG || Max Discount : 8.0 %
,
product name : chair || cost : 780.0 || Manufacturer: LOCAL || Max Discount : 9.0 %
,
product name : chair1 || cost : 780.0 || Manufacturer: LOCAL || Max Discount : 9.0 %
,
product name : Laptop || cost : 80000.0 || Manufacturer: VIVO || Max Discount : 50.0 %
,
product name : Laptop1 || cost : 80000.0 || Manufacturer: VIVO || Max Discount : 25.0 %
]
PS C:\java\oop exp - 07>

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\java\oop exp - 07> javac Product.java Test.java CostSorter.java ManfSorter.java
PS C:\java\oop exp - 07> java Test
Sorting Based on manufacturer :
[
product name : Pen || cost : 10.0 || Manufacturer: CELLO || Max Discount : 3.0 %
,
product name : Pen1 || cost : 10.0 || Manufacturer: CELLO || Max Discount : 3.0 %
,
product name : chair || cost : 780.0 || Manufacturer: LOCAL || Max Discount : 9.0 %
,
product name : chair1 || cost : 780.0 || Manufacturer: LOCAL || Max Discount : 9.0 %
,
product name : Bag || cost : 400.0 || Manufacturer: SKYBAG || Max Discount : 8.0 %
,
product name : Bag1 || cost : 400.0 || Manufacturer: SKYBAG || Max Discount : 8.0 %
,
product name : Laptop || cost : 80000.0 || Manufacturer: VIVO || Max Discount : 50.0 %
,
product name : Laptop1 || cost : 80000.0 || Manufacturer: VIVO || Max Discount : 25.0 %
]
PS C:\java\oop exp - 07>
```

2. Write 1 test case for sorting based on cost Assume that multiple products have same cost.

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  powershell

PS C:\java\oop exp - 07> javac Product.java Test.java CostSorter.java ManfSorter.java
PS C:\java\oop exp - 07> java Test
Sorting Based on Cost if same then on the basis of max discount and if also same then on the basis of product name :
[
product name : Pen || cost : 10.0 || Manufacturer: CELLO || Max Discount : 3.0 %
,
product name : Pen1 || cost : 10.0 || Manufacturer: CELLO || Max Discount : 3.0 %
,
product name : Bag || cost : 400.0 || Manufacturer: SKYBAG || Max Discount : 8.0 %
,
product name : Bag1 || cost : 400.0 || Manufacturer: SKYBAG || Max Discount : 8.0 %
,
product name : chair || cost : 780.0 || Manufacturer: LOCAL || Max Discount : 9.0 %
,
product name : chair1 || cost : 780.0 || Manufacturer: LOCAL || Max Discount : 9.0 %
,
product name : Laptop1 || cost : 80000.0 || Manufacturer: VIVO || Max Discount : 25.0 %
,
product name : Laptop || cost : 80000.0 || Manufacturer: VIVO || Max Discount : 50.0 %
]
PS C:\java\oop exp - 07>
```