

---

## **Practical - 04**

=====

**Student Name** : Vinni Fengade  
**Roll No.** : 67  
**Sem & Sec** : 4 CSE [B]  
**Course Name** : Object Oriented Programming (CSP256)  
**Date Compiled** : 17-June-2022

=====

### **Problem Statements:**

-----

#### **7.Generic and Collection Classes**

A. Write a generic class having a function `addArray()` to add all elements of the array and print the sum. Write a class to test the generic class and its method. Test whether the sum of two arrays is same or not.

B. 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 menu driven 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

---

#### File :Main.java

```
package com.mycompany.practical7;

import java.io.*;
import java.io.BufferedReader;
import java.io.FileReader;
import java.util.ArrayList;
import java.util.Collections;
import java.util.Iterator;
import java.util.Scanner;

public class Practical7 {
    public static void main(String[] args) throws FileNotFoundException,
    IOException {
        //Part A
        System.out.println("-----");
        System.out.println("Part A");
        System.out.println("-----");

        Addition <Integer> ads=new Addition<>();

        Integer[] aa={1,2,3,4,5,6,7,8,9,0};
        ads.addArray(aa);
        //Part A End
        //Part B
        System.out.println("-----");
        System.out.println("Part B");
        System.out.println("-----");

        FileReader f1= new
        FileReader("C:\\Users\\User\\Documents\\NetBeansProjects\\Practical7\\src\\main\\
        \\java\\com\\mycompany\\practical7\\Input\\Product.txt");

        BufferedReader br = new BufferedReader(f1);

        int i=0;
```

---

```

String[] str=new String[4];
String s ,temp;
ArrayList <Product> P=new ArrayList<>(10);

while((s=br.readLine())!=null){
    temp=s;
//    System.out.println(s);
    str = temp.split(" ");
    String name = str[0];
    String M_name=(str[1]);
    int cost = Integer.parseInt(str[2]);
    int discount = Integer.parseInt(str[3]);

    P.add(new Product(name,M_name,cost,discount));
    i++;
}
System.out.println("Buffered Input from File");

System.out.println(P);

System.out.println("++++++++++++++++++++++++++++++++++++++++");
System.out.println("-----");

int choice=1;
//    Iterator Itr = P.iterator();
do{
    Iterator Itr = P.iterator();
    System.out.println("-----
");

    System.out.print("Enter the Input : ");
    Scanner sc=new Scanner(System.in);
    choice =sc.nextInt();
    switch(choice){
        case 1 -> {

```

---

```
        while(Itr.hasNext()){
            Product p=(Product) Itr.next();
            System.out.println(p);
        }
    }
    case 2 -> {
        while(Itr.hasNext()){
            Product p=(Product) Itr.next();
            if(p.getMax_discount()== 50) System.out.println(p);
        }
    }
    case 3 -> {
        Collections.sort(P);
        for(Product pp : P){
            System.out.println(pp);
        }
    }
    case 4 -> {
        Collections.sort(P,new CompareManufacturer());
        for(Product pp : P){
            System.out.println(pp);
        }
    }
    case 0 -> {
        System.out.println("Program terminated");
    }
    default -> System.out.println("Invalid Input");
}
}while(choice!=0);
}
```

---

---

**File : InvalidCountException.java**

```
public class Product implements Comparable<Product>{
    private String product_name, manufacturer;
    private int cost,max_discount;

    Product(String p_name,String m_name,int cost,int discount){
        this.product_name=p_name;
        this.manufacturer = m_name;
        this.max_discount=discount;
        this.cost=cost;
    }

    public String getProduct_name() {
        return product_name;
    }

    public String getManufacturer() {
        return manufacturer;
    }

    public int getCost() {
        return cost;
    }

    public int getMax_discount() {
        this.cost=(this.cost*(100-max_discount))/100;
        return this.max_discount;
    }

    @Override
    public String toString(){
        return ("Product Name :"+this.product_name+"\n Manufacturer Name :
"+this.manufacturer+"\nCost : "+this.cost+"\nMAX Discount :
"+this.max_discount+"\n");
    }
}
```

---

```
@Override
public int compareTo(Product o){
    if(cost<o.cost)
        return 1;
    else if(cost > o.cost)
        return -1;
    else
        return 0;
}
}
```

**File : InvalidTeamName.java**

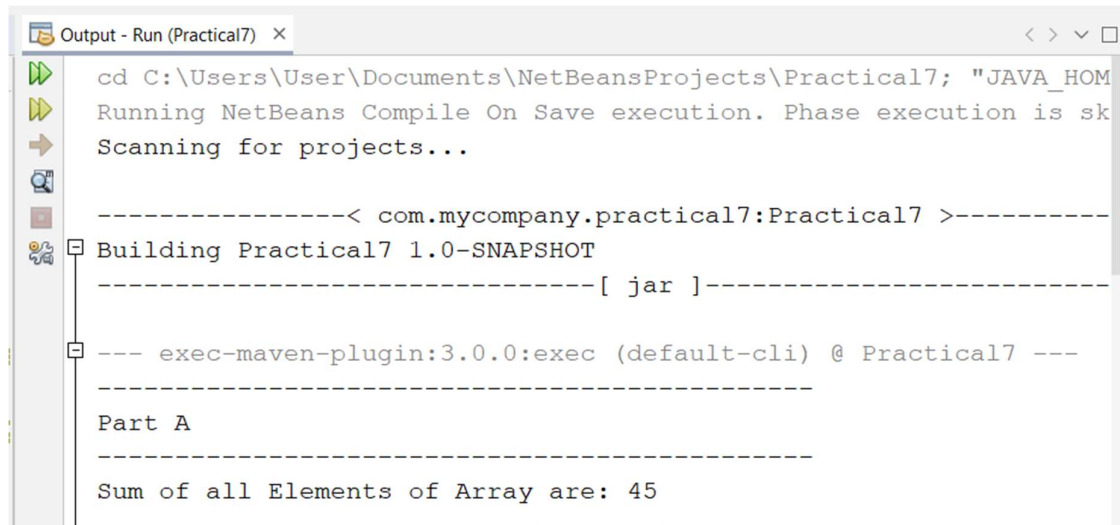
```
package com.mycompany.practical7;
import java.util.*;
import java.util.Comparator;

public class CompareManufacturer implements Comparator<Product>{
    @Override
    public int compare(Product p1, Product p2) {
        return p1.getManufacturer().compareTo(p2.getManufacturer());
    }
}
```

---

## Execution

---



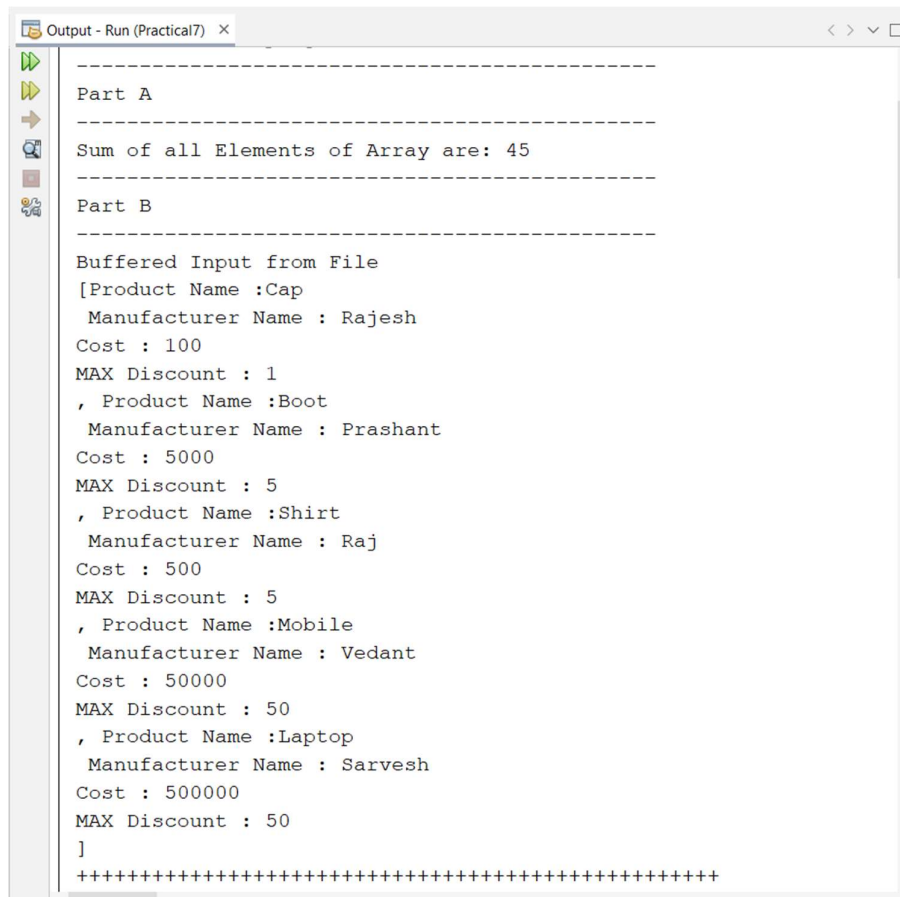
```
Output - Run (Practical7) x
cd C:\Users\User\Documents\NetBeansProjects\Practical7; "JAVA_HOM
Running NetBeans Compile On Save execution. Phase execution is sk
Scanning for projects...

-----< com.mycompany.practical7:Practical7 >-----
Building Practical7 1.0-SNAPSHOT
-----[ jar ]-----

--- exec-maven-plugin:3.0.0:exec (default-cli) @ Practical7 ---

Part A

Sum of all Elements of Array are: 45
```



```
Output - Run (Practical7) x

Part A

Sum of all Elements of Array are: 45

Part B

Buffered Input from File
[Product Name :Cap
Manufacturer Name : Rajesh
Cost : 100
MAX Discount : 1
, Product Name :Boot
Manufacturer Name : Prashant
Cost : 5000
MAX Discount : 5
, Product Name :Shirt
Manufacturer Name : Raj
Cost : 500
MAX Discount : 5
, Product Name :Mobile
Manufacturer Name : Vedant
Cost : 50000
MAX Discount : 50
, Product Name :Laptop
Manufacturer Name : Sarvesh
Cost : 500000
MAX Discount : 50
]
+++++
```

```
Output - Run (Practical7) ×
-----
Enter the Input : 2
Product Name :Mobile
Manufacturer Name : Vedant
Cost : 25000
MAX Discount : 50

Product Name :Laptop
Manufacturer Name : Sarvesh
Cost : 250000
MAX Discount : 50
-----

Output - Run (Practical7) ×
-----
Enter the Input : 3
Product Name :Laptop
Manufacturer Name : Sarvesh
Cost : 250000
MAX Discount : 50

Product Name :Mobile
Manufacturer Name : Vedant
Cost : 25000
MAX Discount : 50

Product Name :Boot
Manufacturer Name : Prashant
Cost : 4750
MAX Discount : 5

Product Name :Shirt
Manufacturer Name : Raj
Cost : 475
MAX Discount : 5

Product Name :Cap
Manufacturer Name : Rajesh
Cost : 99
MAX Discount : 1
-----

Enter the Input : 4
Product Name :Boot
```



```
Output - Run (Practical7) x
-----
Enter the Input : 4
Product Name :Boot
    Manufacturer Name : Prashant
Cost : 4750
MAX Discount : 5

Product Name :Shirt
    Manufacturer Name : Raj
Cost : 475
MAX Discount : 5

Product Name :Cap
    Manufacturer Name : Rajesh
Cost : 99
MAX Discount : 1

Product Name :Laptop
    Manufacturer Name : Sarvesh
Cost : 250000
MAX Discount : 50

Product Name :Mobile
    Manufacturer Name : Vedant
Cost : 25000
MAX Discount : 50

-----
Enter the Input : 0
Program terminated
```