OOPS PRACTICAL - 7

Name : Prathamesh Rajbhoj

Branch : CSE

Roll No : A - 53

Aim :

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

**Question - A**

**Stats.java :**

public class Stats<T extends Number> {

T[] arr;

Stats(T[] array) {

arr = array;

}

public double CalculateSum() {

double sum = 0.0;

for (int i = 0; i < arr.length; i++) {

sum += arr[i].doubleValue();

}

return sum;

}

public boolean sameSum(Stats<?> arr1) {

if (this.CalculateSum() == arr1.CalculateSum())

return true;

else

return false;

}

}

**Test.java :**

public class Test {

public static void main(String[] args) {

Integer[] arr1 = { 1, 2, 3, 4, 5 };

Double[] arr2 = { 1.1, 2.2, 3.3, 4.4, 5.5 };

Integer[] arr3 = { 5, 4, 3, 2, 1 };

Stats<Integer> st1 = new Stats<Integer>(arr1);

Stats<Double> st2 = new Stats<Double>(arr2);

Stats<Integer> st3 = new Stats<Integer>(arr3);

System.out.println(st1.CalculateSum());

System.out.println(st2.CalculateSum());

System.out.println(st3.CalculateSum());

if (st1.sameSum(st3))

System.out.println("SAME");

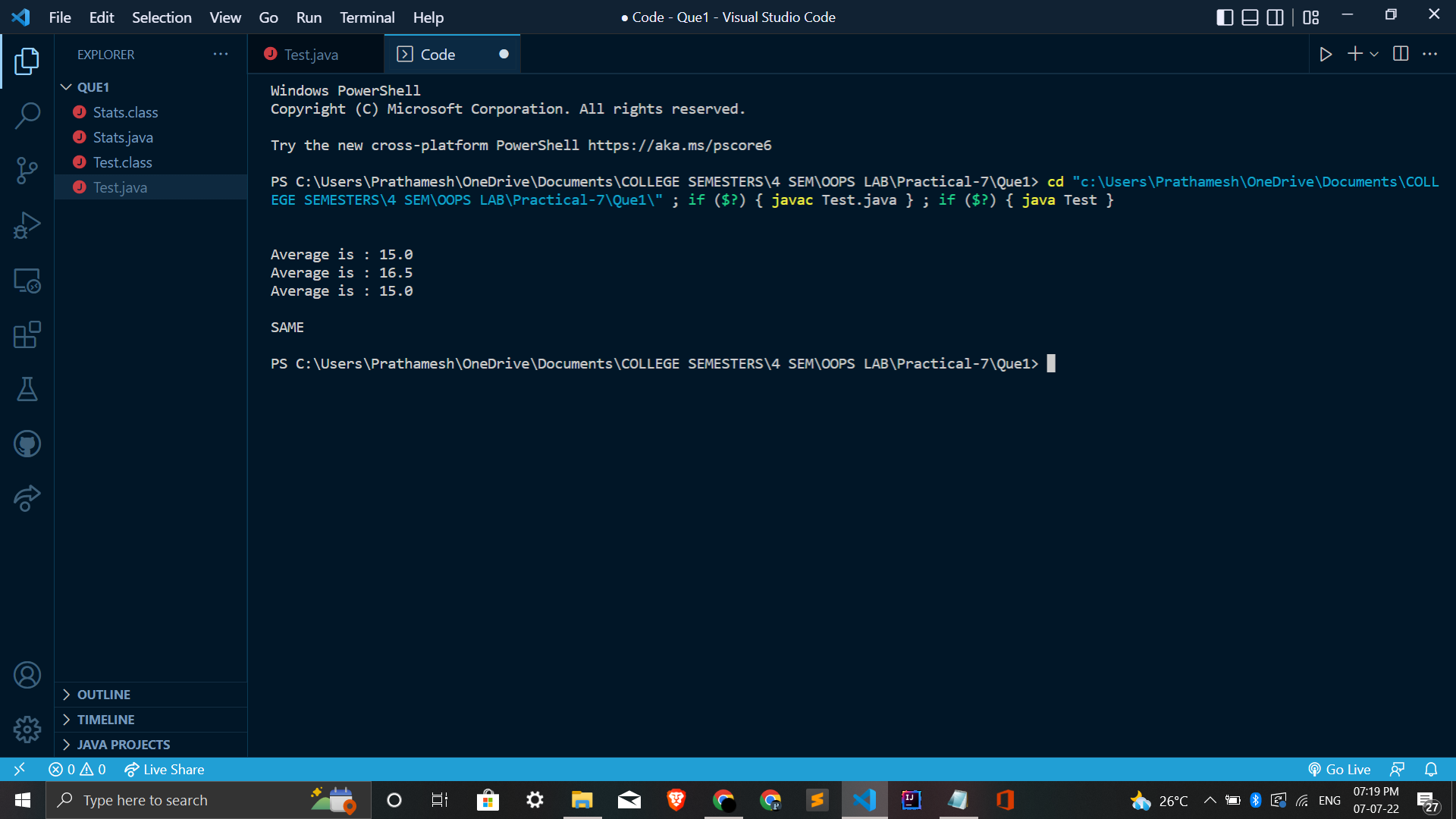
else

System.out.println("DIFFERENT");

}

}

**Output :**



**Question - B**

**Product.java :**

public class Product {

private String product\_name;

private String manufacturer;

private double cost;

private double max\_discount;

Product() {

}

Product(String product\_name, String manufacturer, double cost, double max\_discount) {

this.product\_name = product\_name;

this.manufacturer = manufacturer;

this.cost = cost;

this.max\_discount = max\_discount;

}

double getCost(){

return this.cost;

}

double getDiscount(){

return this.max\_discount;

}

public String toString() {

return this.product\_name + "\t" + this.manufacturer + "\t" + this.cost + "\t" + this.max\_discount;

}

}

**DiscComp .java :**

import java.util.Comparator;

public class DiscComp implements Comparator<Product> {

public int compare(Product p1, Product p2) {

double d1 = p1.getDiscount();

double d2 = p2.getDiscount();

return (d1 < d2 ? -1 : 1);

}

}

**CostComp .java :**

import java.util.Comparator;

public class CostComp implements Comparator<Product> {

public int compare(Product p1, Product p2) {

double d1 = p1.getCost();

double d2 = p2.getCost();

return (d1 < d2 ? -1 : 1);

}

}

**Test.java :**

import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;

import java.util.ArrayList;

import java.util.Collection;

import java.util.Collections;

import java.util.Iterator;

import java.util.Scanner;

import java.util.\*;

public class Test {

public static void main(String[] args) throws IOException {

ArrayList<Product> arr = new ArrayList<Product>();

FileReader fr = null;

BufferedReader br = null;

Scanner sc = new Scanner(System.in);

try {

fr = new FileReader("File.txt");

br = new BufferedReader(fr);

String line;

while ((line = br.readLine()) != null) {

String data[] = line.split(" ");

String name = data[0];

String manuf = data[1];

String st1 = data[2];

String st2 = data[3];

double cost = Double.parseDouble(st1);

double disc = Double.parseDouble(st2);

arr.add(new Product(name, manuf, cost, disc));

}

boolean loop = true;

while (loop) {

System.out.println("\n\nCHOICES AVAILABLE \n");

System.out.println("1. Display Using Iterator");

System.out.println("2. Display the list of products whose max\_discount is 50%");

System.out.println("3. Products sorted according to the cost");

System.out.println("4. Products sorted according to the manufacturer");

System.out.println("5. Exit");

System.out.print("\nEnter Choice : ");

int choice = sc.nextInt();

System.out.println();

switch (choice) {

case 1:

Iterator<Product> it = arr.iterator();

while (it.hasNext()) {

System.out.println(it.next());

}

break;

case 2:

Iterator<Product> itr = arr.iterator();

while (itr.hasNext()) {

Product p1 = itr.next();

if (p1.getDiscount() == 50) {

System.out.print(p1);

System.out.println(" - Final Price : " + 0.5\*p1.getCost());

}

}

break;

case 3:

Collections.sort(arr, new CostComp());

break;

case 4:

Collections.sort(arr, new DiscComp());

break;

case 5:

loop = false;

}

}

} finally {

if (br != null) {

br.close();

}

if (fr != null) {

fr.close();

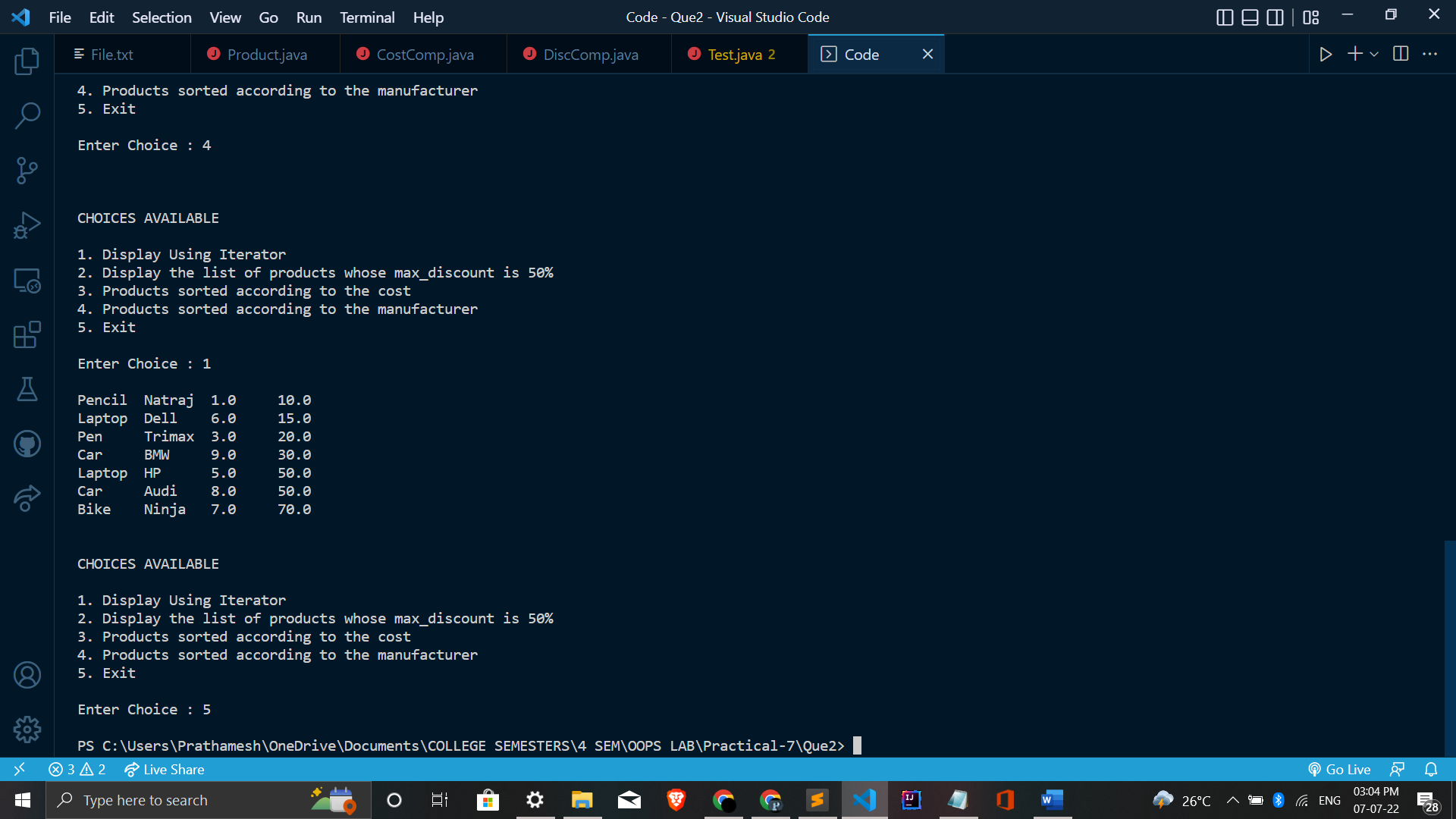
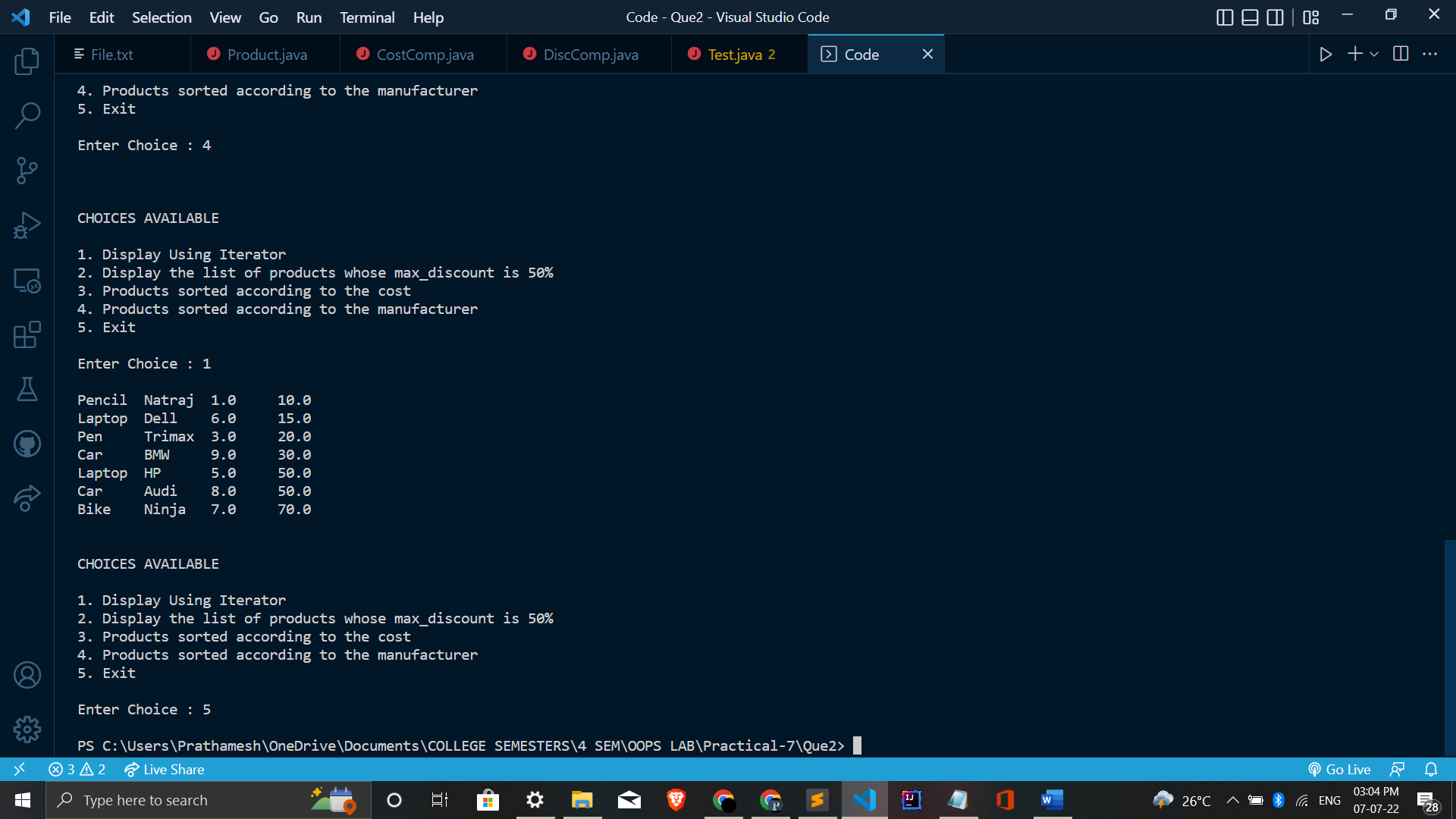
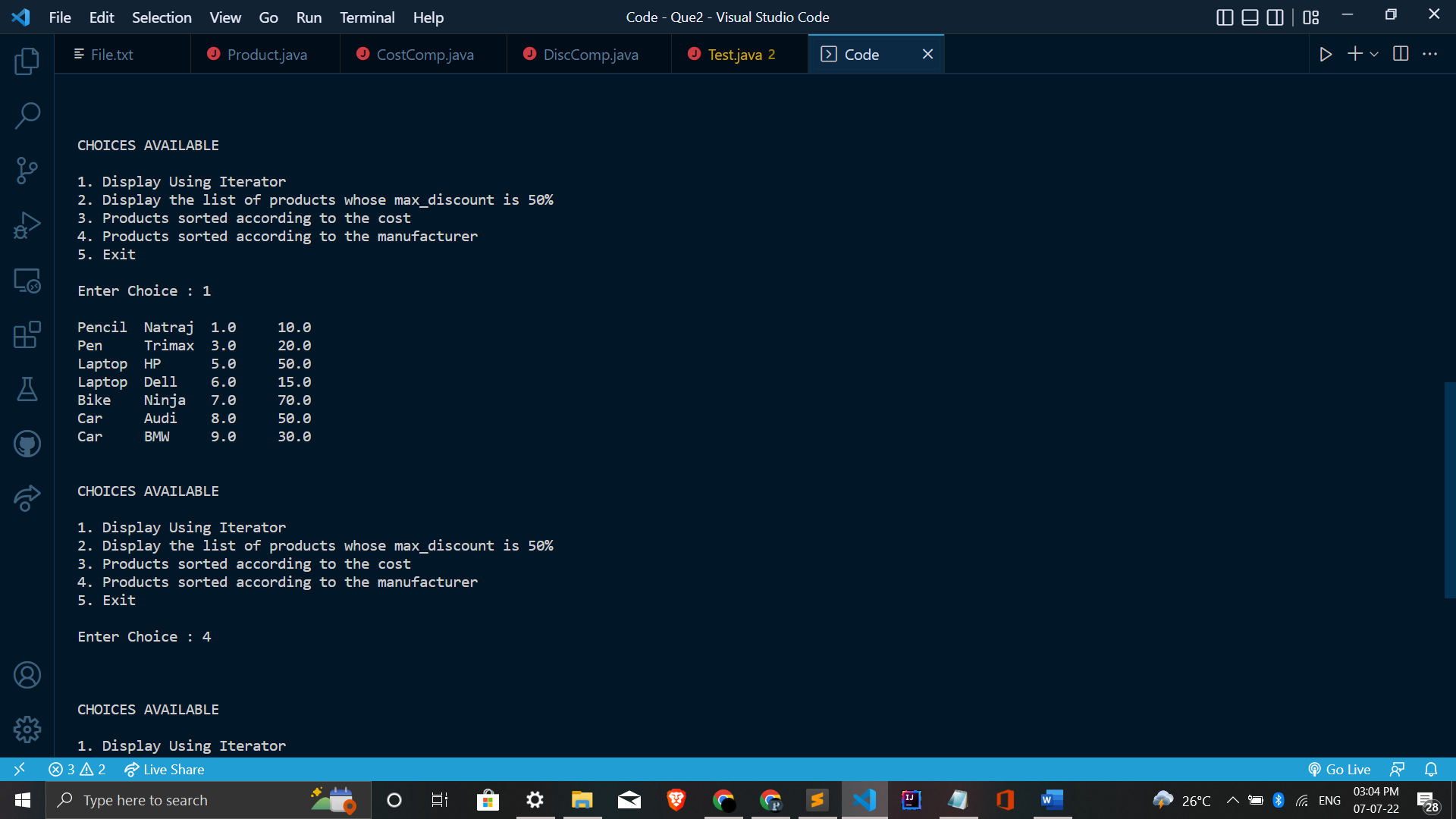
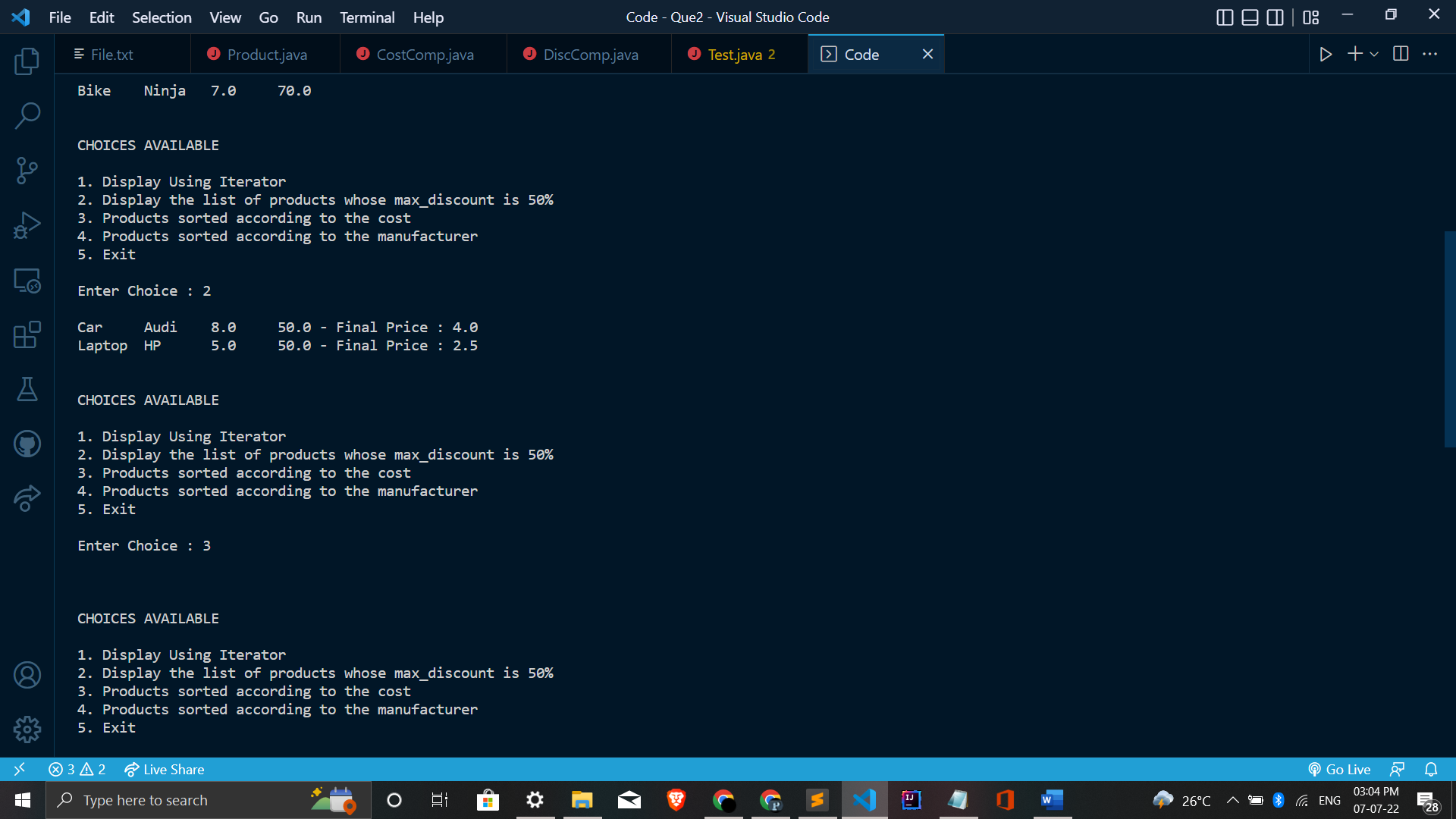
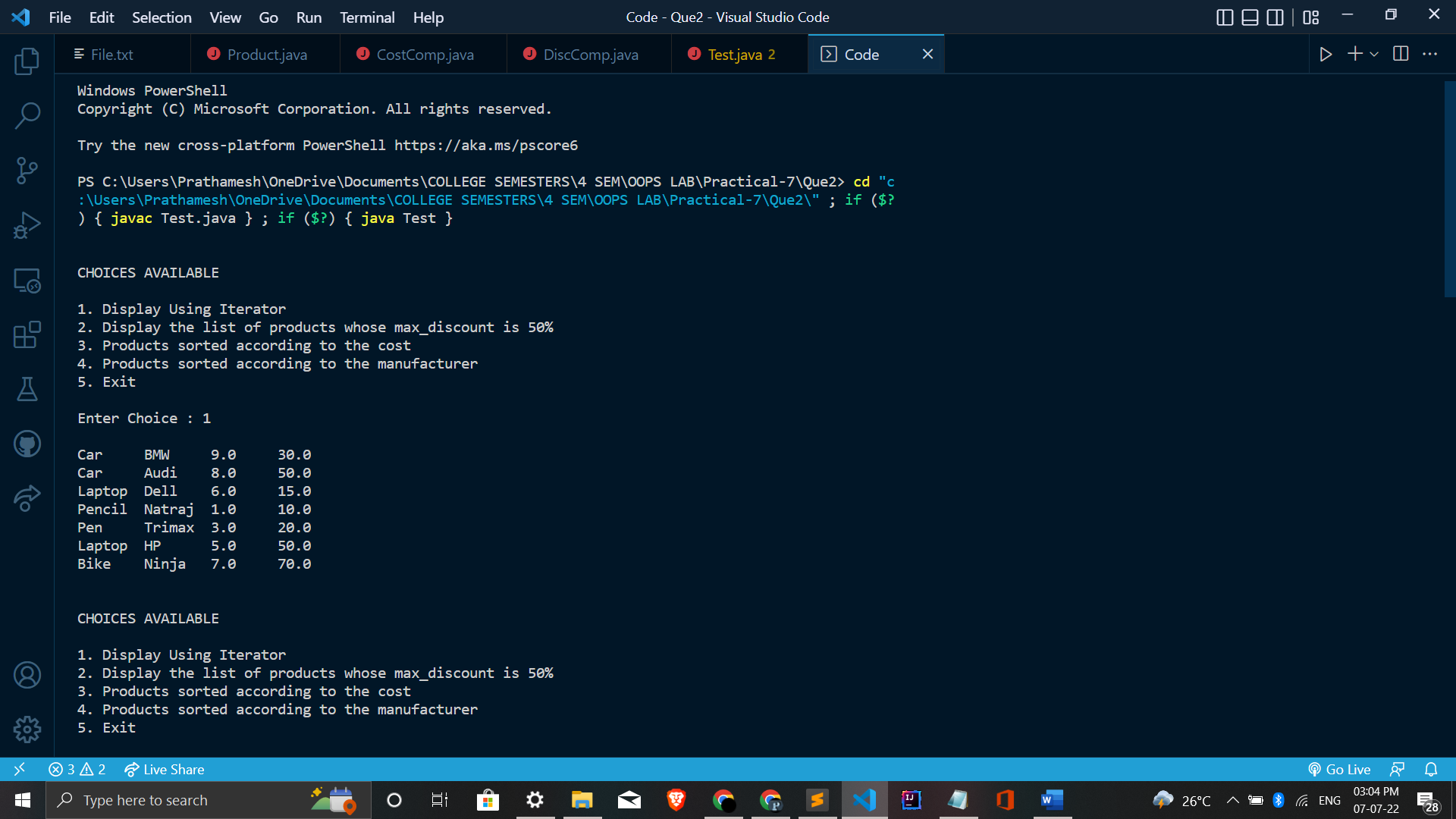
}

sc.close();

}

}

}

**Output :**