**1. Define a class ‘product’ with data members pcode, pname and price. Create 3 objects of**

**the class and find the product having the lowest price.**

**Source Code:**

import java.util.Objects;

public class Product{

String pcode,pname;

int price;

public String getPname(){

return pname;

}

public Product(){

}

public Product(String pname,String pcode,int price){

this.pname=pname;

this.pcode=pcode;

this.price=price;

}

public void setPname(String pname){

this.pname=pname;

}

public String getPcode(){

return pcode;

}

public void setPcode(String pcode){

this.pcode=pcode;

}

public int getPrice(){

return price;

}

public void setPrice(){

this.price=price;

}

public void display(){

System.out.println("pcode :" + this.pcode);

System.out.println("pname :" + this.pname);

System.out.println("price :" + this.price);

}

public static void main(String[] args) {

System.out.println("Name : Thomas V.G\nRollno : 58\nDate : 13/02/2024\nProgram : Symmetric Matrix\n");

Product p1 =new Product();

p1.pcode = "Car123";

p1.pname="Benz";

p1. price=10000;

System.out.println("\*\*\*\*\*\*\*\* Displaying.pl\*\*\*\*\*\*\*...");

p1.display();

Product p2 = new Product("Jaguar", "Car 426", 25000);

System.out.println("\n\*\*\*\*\*\*\*\* Displaying-p2\*\*\*\*\*\*\*\*\*\*");

p2.display();

Product p3 = new Product("Maruti", "Car 800",50000);

System.out.println("\n\*\*\*\*\*\*\*\* Displaying.p3\*\*\*\*\*\*\*\*\*\*");

p3.display();

Product p = p3.getPrice() < (p1.price < p2.price? p1.price:p2.price)? p3: (p1.price < p2.price? p1:p2);

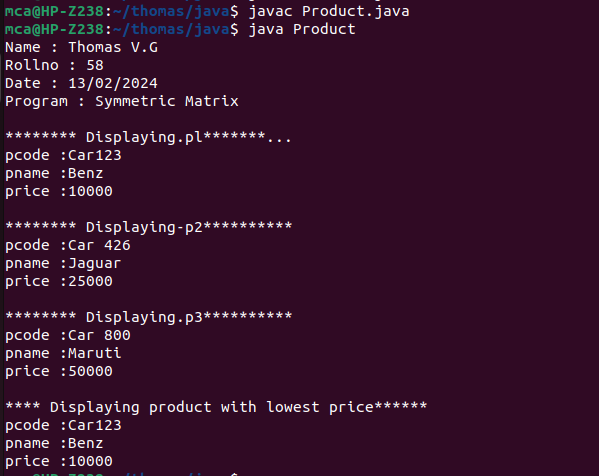
System.out.println("\n\*\*\*\* Displaying product with lowest price\*\*\*\*\*\*");

p.display ();

}

}

**Output**



**2. Read 2 matrices from the console and perform matrix addition.**

**Source Code:**

import java.util.Scanner;

public class MatrixAdd

{

public static void main(String[] args)

{

int p,q,m,n;

Scanner s = new Scanner(System.in);

System.out.print("Enter number of rows in first matrix:");

p = s.nextInt();

System.out.print("Enter number of columns in first matrix:");

q = s.nextInt();

System.out.print("Enter number of rows in second matrix:");

m = s.nextInt();

System.out.print("Enter number of columns in second matrix:");

n = s.nextInt();

if(p==m&&q==n)

{

int a[][] = new int[p][q];

int b[][] = new int[m][n];

int c[][] = new int[m][n];

System.out.println("Enter all the elements of first matrix:");

for(int i=0;i<p;i++)

for(int j=0;j<q;j++)

a[i][j]=s.nextInt();

System.out.println("Enter all the elements of second matrix:");

for(int i=0;i<m;i++)

for(int j=0;j<n;j++)

b[i][j]=s.nextInt();

System.out.println("First Matrix:");

for (int i=0;i<p;i++)

{

for(int j=0;j<q;j++)

System.out.print(a[i][j]+" ");

System.out.println("");

}

System.out.println("Second matrix:");

for(int i=0;i<m;i++)

{

for(int j=0;j<n;j++)

{

System.out.print(b[i][j]+" ");

}

System.out.println(" ");

}

for(int i=0;i<p;i++)

for(int j=0;j<n;j++)

for(int k=0;k<q;k++)

c[i][j]=a[i][j]+b[i][j];

System.out.println("Matrix after addition:");

{

for(int i=0;i<p;i++)

{

for(int j=0;j<n;j++)

System.out.print(c[i][j]+" ");

System.out.println(" ");

}

}

}

else

{

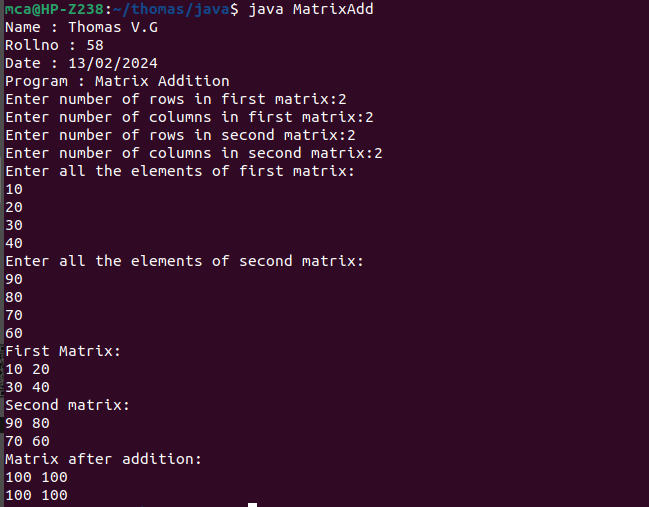
System.out.println("Addition would not be possible");

}

}

}

**Output:**

****

**3. Add complex numbers**

**Source Code:**

public class ComplexNumber{

double real,img;

ComplexNumber(double r,double i)

{

this.real=r;

this.img=i;

}

public static ComplexNumber sum(ComplexNumber c1,ComplexNumber c2)

{

ComplexNumber temp=new ComplexNumber(0,0);

temp.real=c1.real + c2.real;

temp.img=c1.img+c2.img;

return temp;

}

public static void main(String args[])

{

System.out.print("Name : Thomas V.G\nRollno : 58\nDate : 13/02/2024\nProgram : Matrix Addition\n");

ComplexNumber c1=new ComplexNumber(5.5,4);

ComplexNumber c2=new ComplexNumber(1.2,3.5);

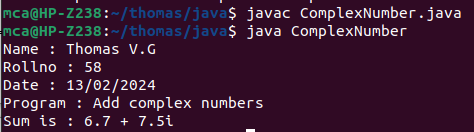
ComplexNumber temp=sum(c1,c2);

System.out.print("Sum is : " + temp.real + " " + " " + temp.img + "i\n");

}

}

**Output:**

****

**4. Read a matrix from the console and check whether it is symmetric or not.**

**Source Code:**

import java.util.Scanner;

public class SymmetricMatrixProgram

{

public static void main(String[] args)

{

System.out.print("Name : Thomas V.G\nRollno : 58\nDate : 13/02/2024\nProgram : Symmetric Matrix\n");

Scanner sc = new Scanner(System.in);

System.out.println("Enter the no. of rows: ");

int rows = sc.nextInt();

System.out.println("Enter the no. of column : ");

int cols = sc.nextInt();

int matrix[][] = new int[rows][cols];

System.out.println("Enter the elements : ");

for(int i=0;i<rows;i++)

for(int j=0;j<cols;j++)

matrix[i][j]=sc.nextInt();

sc.close();

System.out.println("Printing the input matrix:");

for(int i=0;i<rows;i++)

{

for(int j=0;j<cols;j++)

System.out.print(matrix[i][j]+"\t");

System.out.println();

}

if(rows != cols)

System.out.println("The given matrix is not a square matrix.");

else

{

boolean symmetric=true;

for(int i=0;i<rows;i++)

for(int j=0;j<cols;j++)

if(matrix[i][j] != matrix[j][i])

{

symmetric = false;

break;

}

if(symmetric)

{

System.out.println("The given matrix is symmetric");

}

else

{

System.out.println("The given matrix is not symmetric");

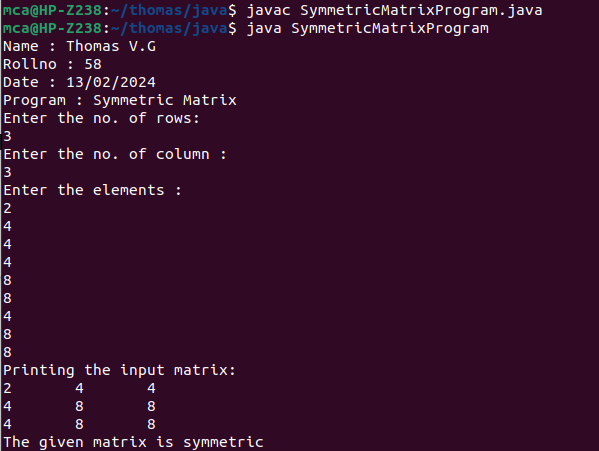
}

}

}

}

**Output:**

****

**5. Create CPU with attribute price. Create inner class Processor (no. of cores, manufacturer) and static nested class RAM (memory, manufacturer). Create an object of CPU and print information of Processor and RAM.**

**Source Code:**

class CPU{

double price;

class Processor{

double cores;

String manufacturer;

double getCache(){

return 4.3;

}

}

protected class RAM{

double memory;

String manufacturer;

double getClockSpeed(){

return 5.5;

}

}

}

public class CPUDetails{

public static void main(String[] args){

System.out.print("Name : Thomas V.G\nRollno : 58\nDate : 13/02/2024\nProgram : CPU Details\n");

CPU cpu = new CPU();

CPU.Processor processor = cpu.new Processor();

CPU.RAM ram = cpu.new RAM();

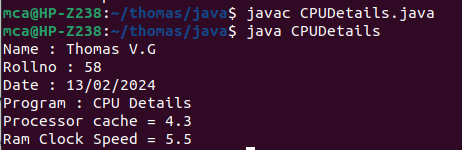
System.out.println("Processor cache = " + processor.getCache());

System.out.println("Ram Clock Speed = "+ ram.getClockSpeed());

}

}

**Output:**

****