**Name: salini k.b**

**Roll No:33**

**Batch:MCA-B**

**Date:06/04/2022**

**OBJECT ORIENTED PROGRAMMING LAB**

**Experiment No.: 1**

**Aim** Define a class “product” with data member pcode,pname and price.Create 3 objects of the class and find the product having the lowest price.

class Product

{

String pcode,pname;

double price;

void details()

{

System.out.println("Product Details");

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

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

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

}

}

public class ProductDetails

{

public static void main (String args[])

{

Product p1 = new Product();

p1.pcode = "THN2134PI";

p1.pname = "REDALME";

p1.price = 11000;

System.out.println("\nProduct1:");

p1.details();

Product p2 = new Product();

p2.pcode = "WECV872WE";

p2.pname = "Bluetooth";

p2.price = 1500;

System.out.println("\nProduct2:");

p2.details();

Product p3 = new Product();

p3.pcode = "QOI43DT6";

p3.pname = "SMART WATCH";

p3.price = 1000;

System.out.println("\nProduct3:");

p3.details();

if(p1.price<p2.price&&p1.price<p3.price)

{

System.out.println("\n\nProduct with lowest price is:");

p1.details();

}

else if(p2.price<p3.price)

{

System.out.println("\n\nProduct with lowest price is:");

p2.details();

}

else

{

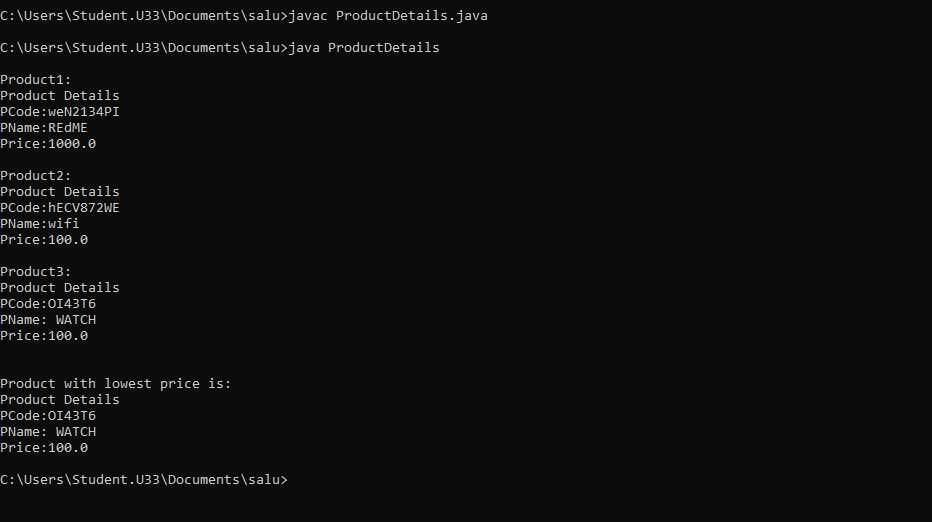
System.out.println("\n\nProduct with lowest price is:");

p3.details();

}

}

}



**Experiment No.: 2**

**Aim**

Read 2 matrices from the console and perform matrix addition.

**Procedure**

import java.util.\*;

class AddMatrix

{

public static void main(String args[])

{

int row, col,i,j;

Scanner in = new Scanner(System.in);

System.out.println("Enter the number of rows");

row = in.nextInt();

System.out.println("Enter the number columns");

col = in.nextInt();

int mat1[][] = new int[row][col];

int mat2[][] = new int[row][col];

int res[][] = new int[row][col];

System.out.println("Enter the elements of matrix1");

for ( i= 0 ; i < row ; i++ )

{

for ( j= 0 ; j < col ;j++ )

mat1[i][j] = in.nextInt();

}

System.out.println("Enter the elements of matrix2");

for ( i= 0 ; i < row ; i++ )

{

for ( j= 0 ; j < col ;j++ )

mat2[i][j] = in.nextInt();

}

for ( i= 0 ; i < row ; i++ )

for ( j= 0 ; j < col ;j++ )

res[i][j] = mat1[i][j] + mat2[i][j] ;

System.out.println("Sum of matrices:-");

for ( i= 0 ; i < row ; i++ )

{

for ( j= 0 ; j < col ;j++ )

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

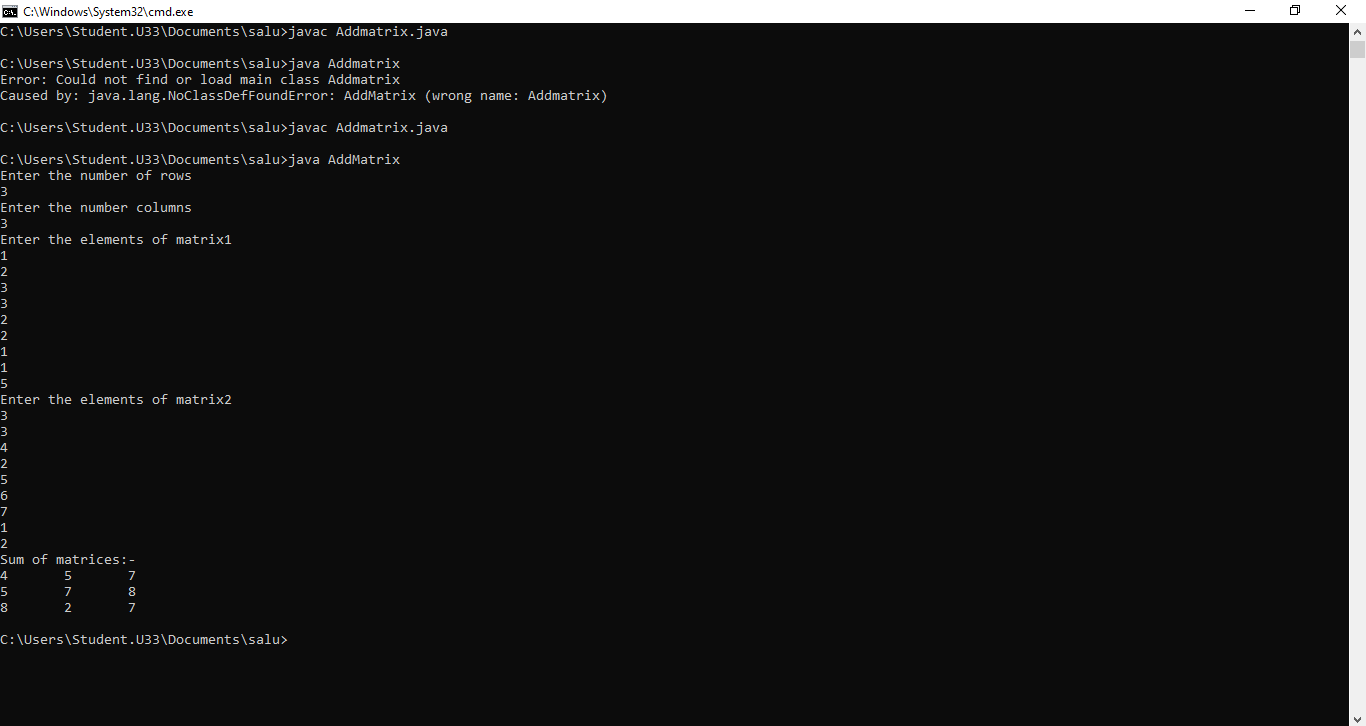
System.out.println();

}

}

}

**Output Screenshot**



**Experiment No.: 3**

**Aim**

Program to add complex numbers

**Procedure**

import java.util.\*;

class Complex {

int real, imaginary;

Complex(){}

Complex(int tempReal, int tempImaginary)

{

real = tempReal;

imaginary = tempImaginary;

}

Complex addComp(Complex C1, Complex C2)

{

Complex temp = new Complex();

temp.real = C1.real + C2.real;

temp.imaginary = C1.imaginary + C2.imaginary;

return temp;

}

Complex subtractComp(Complex C1, Complex C2)

{

Complex temp = new Complex();

temp.real = C1.real - C2.real;

temp.imaginary = C1.imaginary - C2.imaginary;

return temp;

}

void printComplexNumber()

{

System.out.println("Complex number: "

+ real + " + "

+ imaginary + "i");

}}

public class GFG {

public static void main(String[] args)

{

Complex C1 = new Complex(3, 2);

C1.printComplexNumber();

Complex C2 = new Complex(9, 5);

C2.printComplexNumber();

Complex C3 = new Complex();

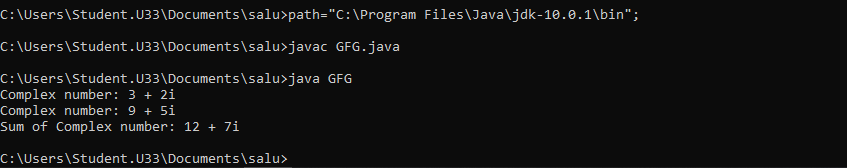
C3 = C3.addComp(C1, C2);

System.out.print("Sum of ");

C3.printComplexNumber();

}}

**Output Screenshot**



**Experiment No.: 4**

**Aim**

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

**Procedure**

import java.util.\*;

public class Symetric {

static void checkSymmetric(int mat[][], int row,int col)

{

int i, j, flag = 1;

System.out.println("The matrix formed is:");

for (i = 0; i < row; i++) {

for (j = 0; j < col; j++) {

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

}

System.out.println("");

}

int[][] transpose = new int[row][col];

for (i = 0; i < row; i++) {

for (j = 0; j < col; j++) {

transpose[j][i] = mat[i][j];

}

}

if (row == col) {

for (i = 0; i < row; i++) {

for (j = 0; j < col; j++) {

if (mat[i][j] != transpose[i][j]) {

flag = 0;

break;

}

}

if (flag == 0) {

System.out.print("\nThe matrix is not symmetric");

break;

}

}

if (flag == 1) {

System.out.print("\nThe matrix is symmetric");

}

}

else {

System.out.print("\nThe matrix is not symmetric");

}

}

public static void main(String args[])

{

Scanner sc = new Scanner(System.in);

int i, j, row, col, flag = 1;

System.out.print("Enter the number of rows:");

row = sc.nextInt();

System.out.print("Enter the number of columns:");

col = sc.nextInt();

int[][] mat = new int[row][col];

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

for (i = 0; i < row; i++) {

for (j = 0; j < col; j++) {

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

}

}

checkSymmetric(mat, row, col);

}

}

**Output Screenshot**

