Experiment : 01

Aim :

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.

CO 1:

Understand object-oriented concepts and design classes and objects to solve problems

Procedure

import java.util.Scanner;

public class Product {

int pcode;

String pname;

int price;

public void ask() {

Scanner cin = new Scanner(System.in);

System.out.println("Enter product code: ");

pcode = cin.nextInt();

System.out.println("Enter product name: ");

pname = cin.next();

System.out.println("Enter product price: ");

price = cin.nextInt();

}

public void printProductId() {

System.out.println("Product code is: " + pcode);

System.out.println("Product name is: " + pname);

System.out.println("Product price is: " + price);

}

public static void main(String[] args) {

Product p1 = new Product();

Product p2 = new Product();

Product p3 = new Product();

p1.ask();

p2.ask();

p3.ask();

p1.printProductId();

p2.printProductId();

p3.printProductId();

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

{

System.out.println(p1.price+"is lowest price");

}

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

{

System.out.println(p2.price+"is lowest price");

}

else{

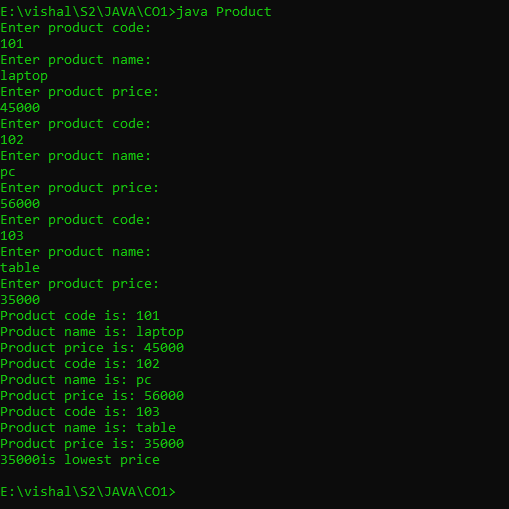
System.out.println(p3.price+"is lowest price");

}

}

}

Output



Experiment : 02

Aim :

Read 2 matrices from the console and perform matrix addition.

CO 1:

Understand object-oriented concepts and design classes and objects to solve problems

Procedure

import java.util.\*;

public class addmatrix{

public static void main(String[] args)

{

int val,n,i,j,r,c;

int arr[][]=new int[50][50];

int arr1[][]=new int[50][50];

Scanner obj=new Scanner(System.in);

System.out.println("enter the size of row :");

r=obj.nextInt();

System.out.println(" enter the size of the column :");

c=obj.nextInt();

System.out.println("enter the elments of first array:");

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

{

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

{

arr[i][j]=obj.nextInt();

}

}

System.out.println("the first array elments are :");

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

{

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

{

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

}

System.out.println();

}

System.out.println("enter the elments of second array :");

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

{

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

{

arr1[i][j]=obj.nextInt();

}

}

System.out.println("the second array elments are :");

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

{

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

{

System.out.print(arr1[i][j]+" ");

}

System.out.println();

}

System.out.println("the sum of array elments is :");

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

{

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

{

System.out.print(arr[i][j]+arr1[i][j]+" ");

}

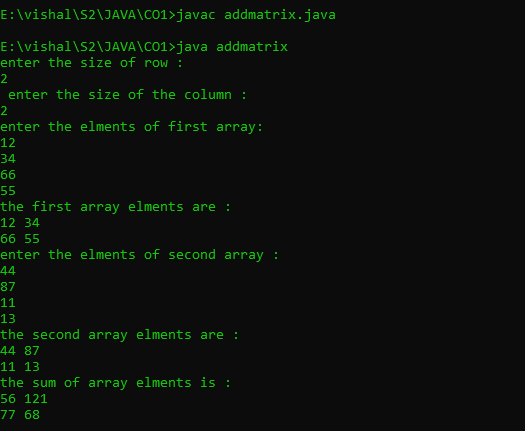
System.out.println();

}

}

}

Output



Experiment : 03

Aim :

Add complex numbers

CO 1:

Understand object-oriented concepts and design classes and objects to solve problems

Procedure

import java.util.\*;

public class complex{

public static void main(String[] args)

{

Scanner obj=new Scanner(System.in);

System.out.println("enter a real number :");

int real=obj.nextInt();

System.out.println("enter imaginary number :");

int img=obj.nextInt();

int comp=real+img;

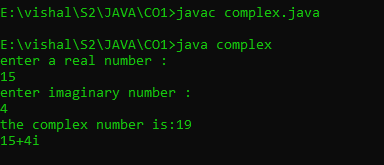
System.out.println("the complex number is:"+comp);

System.out.println(real+"+"+img+ "i " );

}

}

Output



Experiment : 04

Aim :

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

CO 1:

Understand object-oriented concepts and design classes and objects to solve problems

Procedure

import java.util.Scanner;

public class SymtrcMatrx

{

public static void main(String[] args)

{

Scanner obj=new Scanner(System.in);

System.out.println("Enter the row and col of first matrix");

Integer r1=obj.nextInt();

Integer c1=obj.nextInt();

int a[][]=new int[r1][c1];

int c[][]=new int[r1][c1];

int i,j;

System.out.println("enter elements to first array");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

{

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

}

}

System.out.println("Printing 2d first Matrix");

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

{

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

}

System.out.println();

}

int flag=1;

for(i=0;i<r1;i++)

{

for(j=0;j<c1;j++)

{

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

if(c[i][j]!=a[i][j])

{

flag=0;

}

}

}

if(flag==1)

{

System.out.println("The matrix is a Symmetric Matrix");

}

else

{

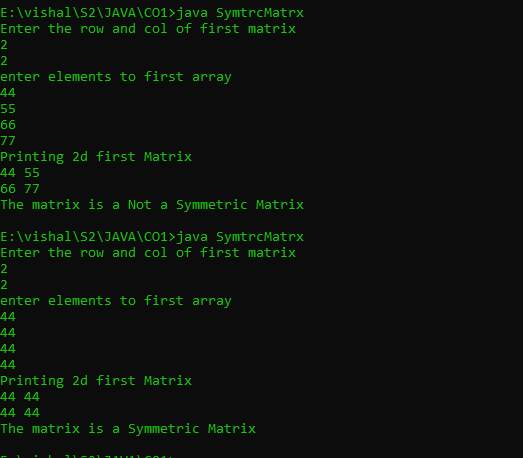
System.out.println("The matrix is a Not a Symmetric Matrix");

}

}

}

Output



Experiment : 05

Aim :

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

CO 1:

Understand object-oriented concepts and design classes and objects to solve problems

Procedure

import java.util.\*;

public class CpuPrgrm {

int price;

public class Processor {

int numCores;

String manufacturer;

Processor(int numCores, String manufacturer) {

this.numCores = numCores;

this.manufacturer = manufacturer;

}

void display() {

System.out.println(" ");

System.out.println("::PROCESSOR INFORMATION::");

System.out.println("Number of cores: " + numCores);

System.out.println("Manufacturer: " + manufacturer);

System.out.println(" ");

}

}

static class RAM {

int memory;

String manufacturer;

RAM(int memory, String manufacturer) {

this.memory = memory;

this.manufacturer = manufacturer;

}

void display() {

System.out.println(" ");

System.out.println("::RAM INFORMATION::");

System.out.println("Memory: " + memory);

System.out.println("Manufacturer: " + manufacturer);

System.out.println(" ");

}

}

public static void main(String[] args) {

Scanner Snr = new Scanner(System.in);

CpuPrgrm cpu = new CpuPrgrm();

System.out.print("Enter CPU price: ");

cpu.price = Snr.nextInt();

System.out.print("Enter number of processor cores: ");

int numCores = Snr.nextInt();

System.out.print("Enter processor manufacturer: ");

String processorManufacturer = Snr.next();

CpuPrgrm.Processor processor = cpu.new Processor(numCores, processorManufacturer);

processor.display();

System.out.print("Enter RAM memory: ");

int ramMemory = Snr.nextInt();

System.out.print("Enter RAM manufacturer: ");

String ramManufacturer = Snr.next();

CpuPrgrm.RAM ram = new CpuPrgrm.RAM(ramMemory, ramManufacturer);

ram.display();

}

}

Output

