**JAVA LAB CYCLE-1**

1. Java Program to Accept the Marks of a Student into a 1D Array and find total marks and percentage

import java.util.Scanner;

public class percentage

{

public static void main(String[] args)

{

int n, count = 0, total = 0, percentage;

Scanner s = new Scanner(System.in);

System.out.print("Enter no. of subject:");

n = s.nextInt();

int a[] = new int[n];

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

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

{

a[i] = s.nextInt();

total = total + a[i];

}

percentage = total / n;

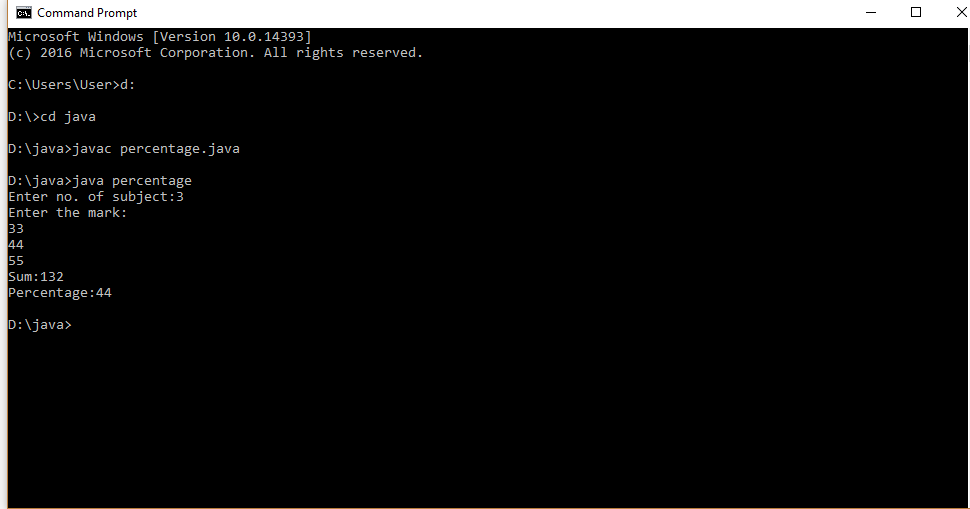
System.out.println("Sum:"+total);

System.out.println("Percentage:"+percentage);

}

}

OUTPUT



1. Java Program to Count the Number of Occurrence of an Element in an Array

import java.util.Scanner;

public class CountOccurrence

{

public static void main(String[] args)

{

int n, x, count = 0, i = 0;

Scanner s = new Scanner(System.in);

System.out.print("Enter no. of elements you want in array:");

n = s.nextInt();

int a[] = new int[n];

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

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

{

a[i] = s.nextInt();

}

System.out.print("Enter the element of which you want to count number of occurrences:");

x = s.nextInt();

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

{

if(a[i] == x)

{

count++;

}

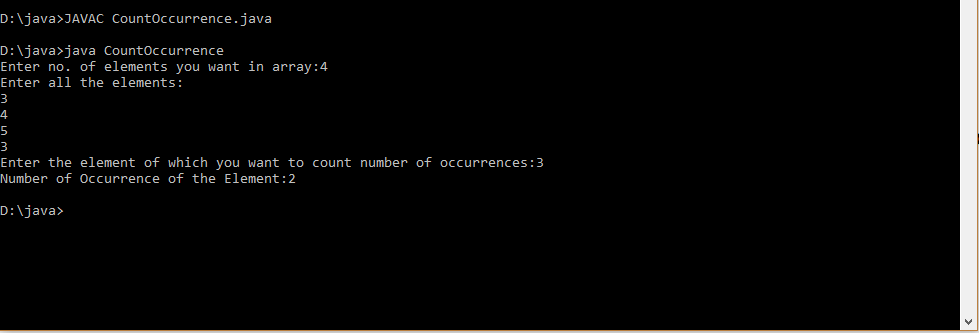
}

System.out.println("Number of Occurrence of the Element:"+count);

}

}

OUTPUT



1. Java Program to Add Two MXN Matrix from User Input

import java.util.Scanner;

public class multi

{

public static void main(String[] args)

{

int p, q;

Scanner s = new Scanner(System.in);

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

p = s.nextInt();

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

q = s.nextInt();

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

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

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

System.out.println("Enter all the elements of 1 st matrix:");

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

{

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

{

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

}

}

System.out.println("The Matrix is:");

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("Enter all the elements of 2 nd matrix:");

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

{

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

{

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

}

}

System.out.println("The Matrix is:");

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

{

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

{

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

}

System.out.println("");

}

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

{

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

{

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

}

}

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

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

{

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

{

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

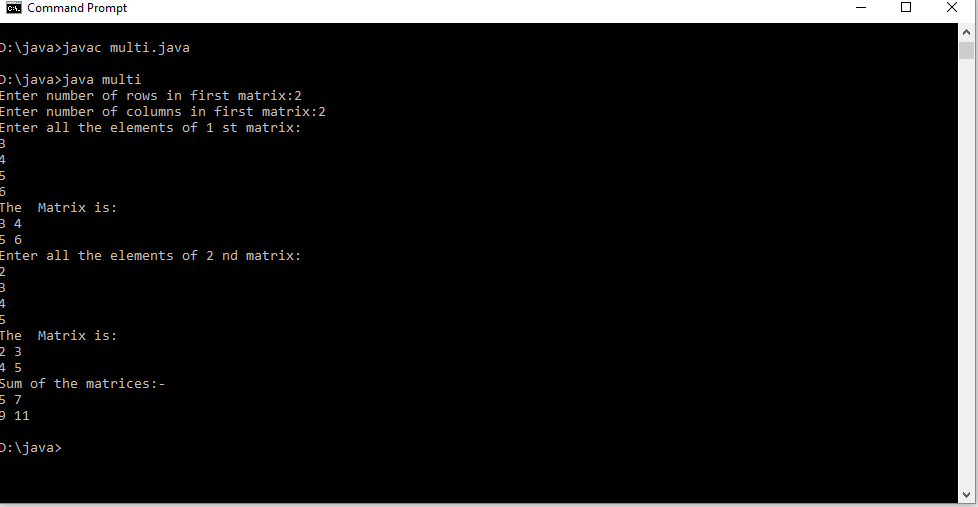
}

System.out.println("");

}

}}

OUTPUT



1. Add complex numbers

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;

}

}

public class GFG

{

public static void main(String[] args)

{

Complex C1 = new Complex(3, 2);

System.out.println("Complex number 1 : "

+ C1.real + " + i"

+ C1.imaginary);

Complex C2 = new Complex(9, 5);

System.out.println("Complex number 2 : "

+ C2.real + " + i"

+ C2.imaginary);

Complex C3 = new Complex();

C3 = C3.addComp(C1, C2);

System.out.println("Sum of complex number : "

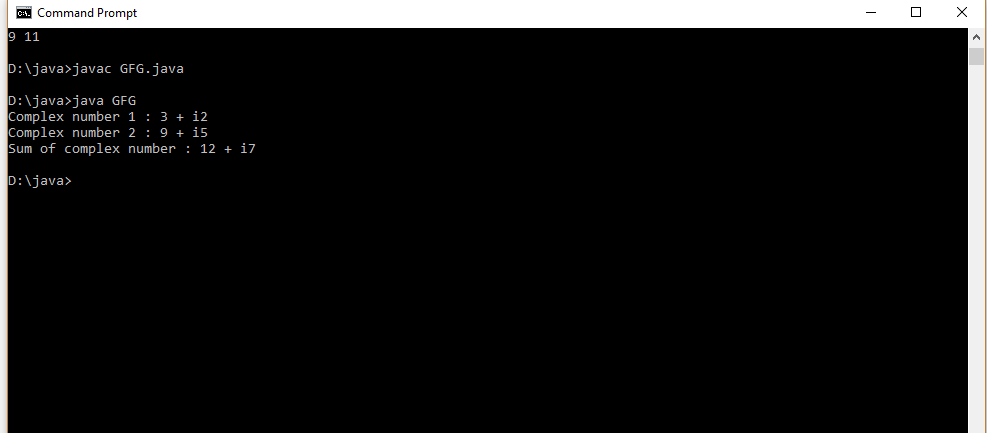
+ C3.real + " + i"

+ C3.imaginary);

}

}

OUTPUT



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

import java.util.Scanner;

public class Symmetric

{

public static void main(String[] args)

{

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 columns : ");

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();

}

}

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, so it can't be symmetric.");

}

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...");

}

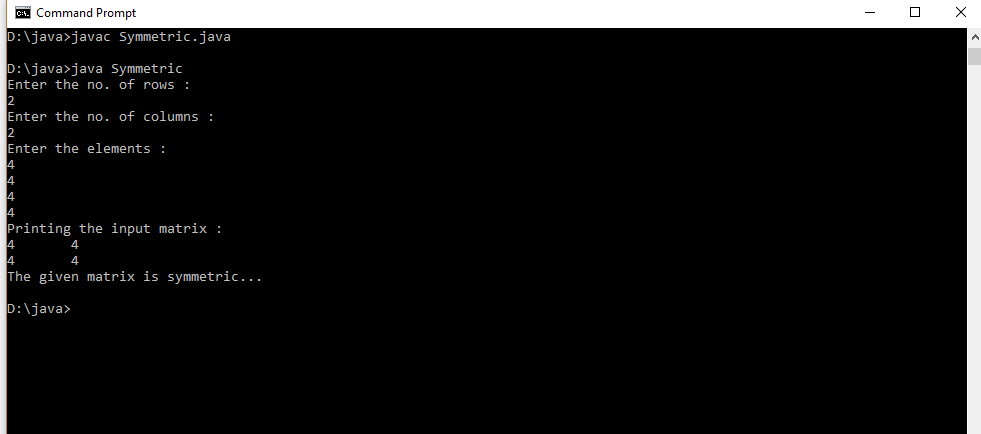
}

sc.close();

}

}

OUTPUT



6.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.

import java.util.\*;

public class Products

{

int pcode;

String pname;

int price;

Products()

{

pcode=0;

pname=null;

price=0;

}

public static void cal(int a,int b,int c)

{

int p1=a;

int p2=b;

int p3=c;

float lowest;

if(p1<p2)

{

if(p3<p1)

{

lowest = p3;

}

else

{

lowest = p1;

}

}

else

{

if(p2<p3)

{

lowest = p2;

}

else

{

lowest = p3;

}

}

System.out.println("The lowest price among the 3 Product is : "+lowest);

}

public static void main(String args[])

{

Scanner s = new Scanner(System.in);

Products p1= new Products();

Products p2= new Products();

Products p3= new Products();

System.out.print("Enter the Product 1 Code : ");

p1.pcode =s.nextInt();

System.out.print("Enter the Product 1 Name : ");

p1.pname = s.next();

System.out.print("Enter the Product 1 Price : ");

p1.price=s.nextInt();

System.out.print("Enter the product 2 Code : ");

p2.pcode =s.nextInt();

System.out.print("Enter the Product 2 Name : ");

p2.pname = s.next();

System.out.print("Enter the Product 2 Price : ");

p2.price = s.nextInt();

System.out.print("Enter the Product 3 Code : ");

p3.pcode =s.nextInt();

System.out.print("Enter the Product 3 Name : ");

p3.pname = s.next();

System.out.print("Enter the Product 3 Price : ");

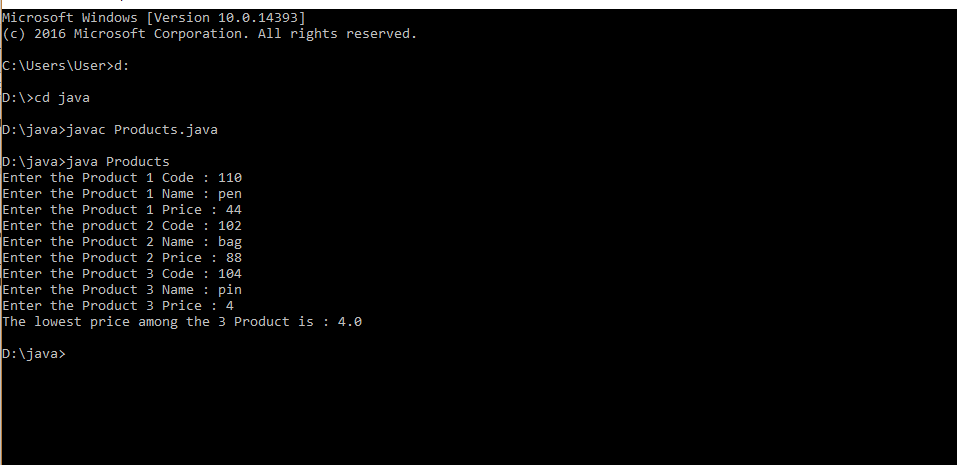
p3.price = s.nextInt();

Products.cal(p1.price,p2.price,p3.price);

}

}

Output



7.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. (---No need to do now---)

class CPU

{

double price;

// nested class

class Processor{

double cores;

String manufacturer;

double getCache()

{

return 4.3;

}

}

// nested protected class

protected class RAM{

double memory;

String manufacturer;

double getClockSpeed(){

return 5.5;

}

}

}

public class Main

{

public static void main(String[] args)

{

CPU cpu = new CPU();

// object of inner class Processor

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

// object of inner class RAM

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

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

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

}

}

OUTPUT

