**OBJECT ORIENTED PROGRAMMING LAB**

**Jomin K Mathew**

**TKM20MCA2021**

**Roll.no: 20MCA221**

**Program 1**

29/04/2021

**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.

**Algorithm:**

Step 1: Start.

Step 2: Create a class having name Products and members as

pcode, pname and price.

Step 3: Declare three objects in the class and add the values of

each data members into objects.

Step 4: Using if condition check which object has the lowest

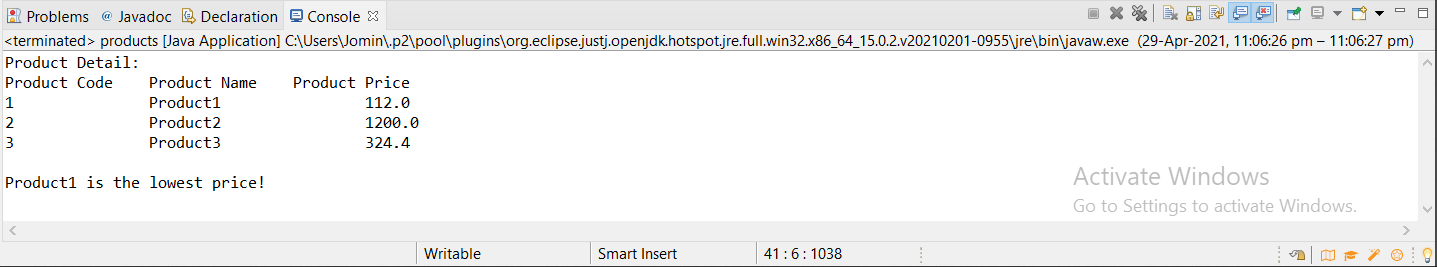
price and print it.

Step 5: Stop.

**Source Code:**

|  |  |
| --- | --- |
|  | **package** packoops;  **public** **class** products {  **int** pcode;  String pname;  **double** price;  **double** lowest;  **void** data(**int** c, String n, **double** p){  pcode=c;  pname=n;  price=p;  }  **void** display(){  System.***out***.println(pcode+"\t\t"+pname+"\t\t"+price);  }  **static** **void** lowest(**double** price1,**double** price2, **double** price3){  **if**(price1<=price2 && price1<=price3){  System.***out***.println("\nProduct1 is the lowest price!");  }  **else** **if**(price2<=price1 && price2<=price3){  System.***out***.println("\nProduct2 is the lowest price!");  }  **else**{  System.***out***.println("\nProduct3 is the lowest price!");  }  }  **public** **static** **void** main(String[] args){  products obj1 = **new** products();  products obj2 = **new** products();  products obj3 = **new** products();  obj1.data(1,"Product1",112);  obj2.data(2,"Product2",1200);  obj3.data(3,"Product3",324.4);  System.***out***.println("Product Detail:\nProduct Code\tProduct Name\tProduct Price");  obj1.display();  obj2.display();  obj3.display();  *lowest*(obj1.price,obj2.price,obj3.price);  }  } |

# Output :



**Program 2**

29/04/2021

**Aim:** Read 2 matrices from the console and perform matrix

addition.

**Algorithm:**

Step 1: Start

Step 2: Create a class matrix.

Step 3: Take input values from user(order of two)

Step 4: If both matrix have equal order then Perform addition

operation of matrix

Step 5: Addition operation is done then print the resultant

matrix

Step 6: Stop

**Source Code:**

**package** packoops;

**import** java.util.\*;

**public** **class** matrix {

**int** row;

**int** column;

**int**[][] array = **new** **int**[10][10];

**public** **void** get\_metrix(){

**int** rc,cc;

Scanner sc= **new** Scanner(System.***in***);

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

**this**.row = sc.nextInt();

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

**this**.column = sc.nextInt();

System.***out***.print("Enter matrix elements : ");

**for**(rc=0;rc<**this**.row;rc++){

**for**(cc=0;cc<**this**.column;cc++){

**this**.array[rc][cc] = sc.nextInt();

}

}

}

**public** **static** matrix sum(matrix c1, matrix c2) {

**int** rc, cc;

matrix temp = **new** matrix();

**if** (c1.row == c2.row && c1.column == c2.column) {

temp.row =c1.row;

temp.column = c1.column;

**for** (rc = 0; rc < c1.row; rc++) {

**for** (cc = 0; cc < c1.column; cc++) {

temp.array[rc][cc] = c1.array[rc][cc] + c2.array[rc][cc];

}

}

}

**else** {

System.***out***.println("Order of matrixs is not same ");

}

**return** temp;

}

**public** **void** display\_matrix(){

**int** rc,cc;

**for**(rc=0;rc<**this**.row;rc++){

**for**(cc=0;cc<**this**.column;cc++){

System.***out***.print(**this**.array[rc][cc] + "\t" );

}

System.***out***.println("");

}

}

**public** **static** **void** main(String[] args) {

matrix first = **new** matrix();

matrix second = **new** matrix();

matrix temp = *sum*(first, second);

first.get\_metrix();

second.get\_metrix();

temp = *sum*(first,second);

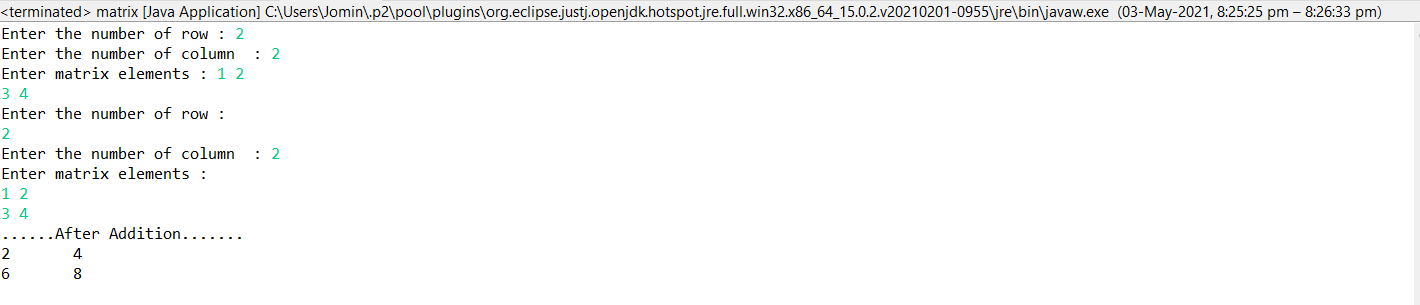
System.***out***.println("......After Addition.......");

temp.display\_matrix();

}

}

**Output:**

****

**Program 3**

03/05/2021

**Aim:** Add complex numbers.

**Algorithm:**

Step 1: Start

Step 2: Create a class complexadd.

Step 3: Take input values from user

Step 4: Add corresponding values

Step 5: Addition operation is done then print the resultant

complex number

Step 6: Stop

**Source Code:**

**package** packoops;

**public** **class** complexadd {

**double** real, img;

complexadd(**double** r, **double** i){

**this**.real = r;

**this**.img = i;

}

**public** **static** complexadd sum(complexadd c1, complexadd c2)

{

complexadd temp = **new** complexadd(0, 0);

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

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

**return** temp;

}

**public** **static** **void** main(String args[]) {

complexadd c1 = **new** complexadd(7.5, 2);

complexadd c2 = **new** complexadd(1.2, 3.5);

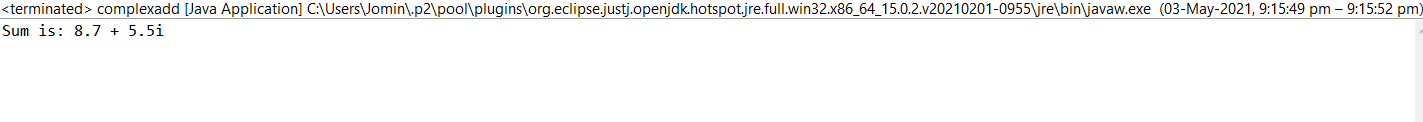
complexadd temp = *sum*(c1, c2);

System.***out***.printf("Sum is: "+ temp.real+" + "+ temp.img +"i");

}

}

**Output:**

****

**Program 4**

03/05/2021

**Aim:** Read a matrix from the console and check whether it is

symmetric or not.

**Algorithm:**

Step 1: Start

Step 2: Create a class symmetricmatrix.

Step 3: Take input values from user.

Step 4: check whether the matrix is symmetrix or not.

Step 5: if it is a symmetrix then print symmetric otherwise

print not a synnmetric matrix.

Step 6: Stop

**Source Code:**

**package** packoops;

**import** java.util.Scanner;

**public** **class** symmetricmatrix {

**public** **static** **void** main(String[] args) {

**int** x=0;

Scanner scan=**new** Scanner(System.***in***);

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

**int** row=scan.nextInt();

System.***out***.println("Enter the number of columns in matrix");

**int** col=scan.nextInt();

**if**(row!=col) {

System.***out***.println("Cannot find Symmetry for this matrix");

}

**else** {

**int**[][] a=**new** **int**[10][10];

System.***out***.println("Enter the values of matrix");

{

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

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

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

}

}

}

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

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

**if**(a[i][j]!=a[j][i]) {

x=x+1;

**break**;

}

**if**(x>0) {

**break**;

}

}

}

**if**(x==0) {

System.***out***.println("It is a Symmetric Matrix");

}

**else** {

System.***out***.println("It is not a Symmetric Matrix");

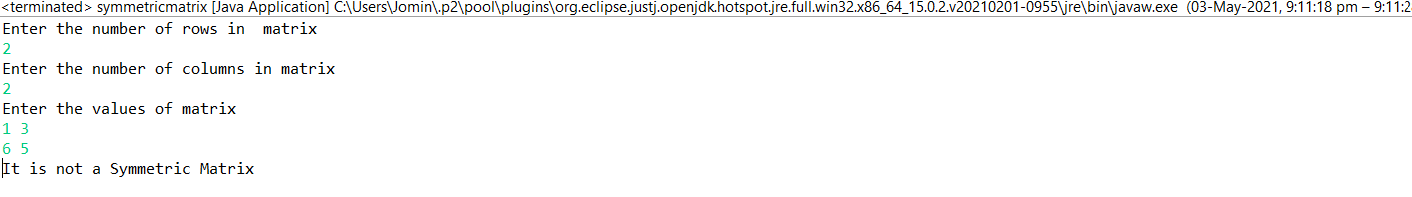
}

}

}

}

**Output:**

****

**Program 5**

06/05/2021

**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.

**Algorithm:**

Step 1: Start.

Step 2: Create a class cpu with data member price and class

processor.

Step 3: Class processor contain data members

no\_of\_core, manufacturer and a nested class RAM.

Step 4: class RAM contain memory and manufacturer as data

members.

Step 5: Create objects in corresponding classes and display it’s

details.

Step 6: Stop

**Source Code:**

**package** packoops;

**public** **class** cpu {

**int** price;

**public** **class** processor

{

**int** no\_of\_core;

String manufacturer;

}

**static** **class** RAM

{

**int** memory;

String manufacturer;

}

**public** **void** display(**int** x,**int** y,**int** z,String g, String f)

{ System.***out***.println("\n Processor");

System.***out***.println("\n Price-> "+x+"\n"+"Number of cores-> "+y+"\n"+"Manufacturer-> "+z);

System.***out***.println("\n RAM ");

System.***out***.println(" \n Memory: "+g+"\n"+"Manufacturer: "+z);

}

**public** **static** **void** main(String[] args) {

**int** x,y,z;

String g,f;

cpu obj=**new** cpu();

cpu.processor obj1=obj.**new** processor();

x=obj.price=17964;

y=obj1.no\_of\_core=6;

g=obj1.manufacturer="Ryzen";

cpu.RAM obj2=**new** RAM();

z=obj2.memory=16;

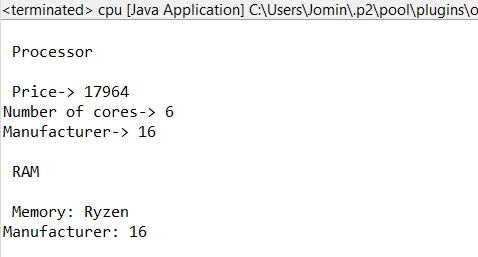
f=obj2.manufacturer="Crucial";

obj.display(x,y,z,g,f);

}

}

**Output:**

****

**Program 6**

06/05/2021

**Aim:** Program to Sort strings.

**Algorithm:**

Step 1: Start

Step 2: Create a class stringsort

Step 3: Take input values from user

Step 4: Check each string

Step 5: Print sorted order strings

Step 6: Stop

**Source Code:**

**package** packoops;

**import** java.util.Scanner;

**public** **class** stringsort {

**public** **static** **void** main(String[] args) {

**int** count;

String string;

Scanner s=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of strings: ");

count=s.nextInt();

String str\_arr[]=**new** String[count];

Scanner sc=**new** Scanner(System.***in***);

System.***out***.println("Enter the strings: ");

**for**(**int** i=0;i<count;i++)

{

str\_arr[i]=sc.nextLine();

}

s.close();

sc.close();

**for**(**int** i=0;i<count;i++)

{

**for**(**int** j=i+1;j<count;j++)

{

**if**(str\_arr[i].compareTo(str\_arr[j])>0)

{

string=str\_arr[i];

str\_arr[i]=str\_arr[j];

str\_arr[j]=string;

}

}

}

System.***out***.println("String after sorting: ");

**for**(**int** i=0;i<count;i++)

{

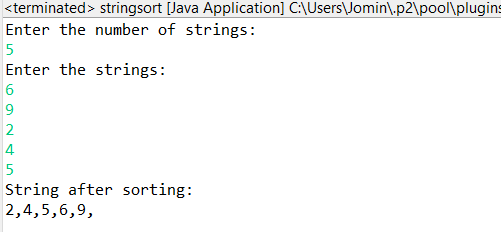
System.***out***.print(str\_arr[i]+",");

}

}

}

**Output:**

****

**Program 7**

07/05/2021

**Aim:** Search an element in an array.

**Algorithm:**

Step 1: Start

Step 2: Create a class searchelement.

Step 3: Take input values from user to an array

Step 4: Check each element in the array

Step 5: if the searched element is present print Element found

else print Element not found.

Step 6: Stop

**Source Code:**

**package** packoops;

**import** java.util.Scanner;

**public** **class** searchelement

{

**public** **static** **void** main(String[] args)

{

**int** n, x, flag = 0, i = 0;

Scanner s = **new** Scanner(System.***in***);

System.***out***.print("Enter number of elements:");

n = s.nextInt();

**int** a[] = **new** **int**[n];

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

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

{

a[i] = s.nextInt();

}

System.***out***.print("Enter the element to be searched:");

x = s.nextInt();

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

{

**if**(a[i] == x)

{

flag = 1;

**break**;

}

**else**

{

flag = 0;

}

}

**if**(flag == 1)

{

System.***out***.println("Element found");

}

**else**

{

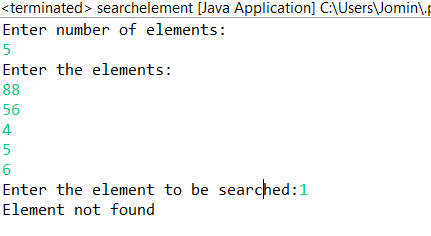
System.***out***.println("Element not found");

}

}

}

**Output:**



**Program 8**

10/05/2021

**Aim:** Perform string manipulations

**Algorithm:**

Step 1: Start

Step 2: Create a class stringhandling

Step 2: Get the strings from the user

Step 3: Display the length of the first string

Step 4: Display the length of the second string

Step 5: Display the combined string with lowercase

Step 6: Display the second string with uppercase

Step 7: Display the combined String after replacing characters

Step 8: Stop

**Source Code:**

**package** packoops;

**import** java.util.Scanner;

**public** **class** stringhandling {

**public** **static** **void** main(String[] args)

{

Scanner s=**new** Scanner(System.***in***);

System.***out***.println("Enter the first strings: ");

String s1=s.nextLine();

System.***out***.println("Enter the Second strings: ");

String s2=s.nextLine();

System.***out***.println("Enter the Third strings: ");

String s3=s.nextLine();

System.***out***.println("Enter the Fourth strings: ");

String s4=s.nextLine(); ;

System.***out***.println("String 1:"+s1);

System.***out***.println("String 2:"+s2);

System.***out***.println("String concatenation:"+s1.concat(s2));

System.***out***.println("String length of first string:"+s1.length());

System.***out***.println("String comparison of two strings:"+s1.compareTo(s2));

System.***out***.println("String 2 empty or not:"+s2.isEmpty());

System.***out***.println("String 3 empty or not:"+s3.isEmpty());

System.***out***.println("Before trimming:"+s4);

System.***out***.println("After String trim:"+s4.trim());

System.***out***.println("String toLowerCase():"+s1.toLowerCase());

System.***out***.println("String toUpperCase():"+s2.toUpperCase());

System.***out***.println("String replace():"+(s1.replace("H", "B")));

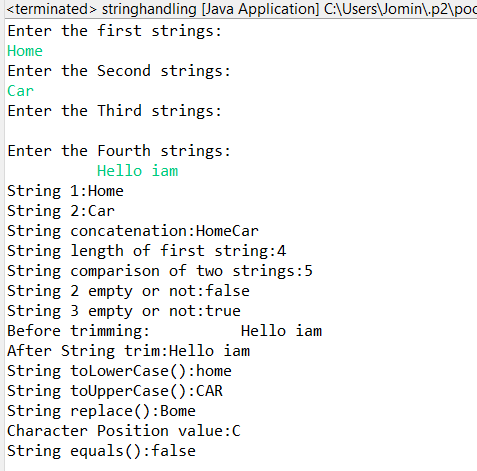
System.***out***.println("Character Position value:"+s2.charAt(0));

System.***out***.println("String equals():"+s1.equals(s2));

}

}

**Output:**



**Program 9**

20/05/2021

**Aim:** Program to create a class for Employee having

attributes eNo, eName eSalary. Read n employ

information and Search for an employee given eNo,

using the concept of Array of Objects.

**Algorithm:**

Step 1: Start

Step 2: Create a class employee with members eNo, eName,

Salary.

Step 2: Enter the number of employees from the user

Step 3: Enter the details of employee

Step 4: Enter the eNo to be searched

Step 5: Display the detail of that employee

Step 6: Stop

**Source Code:**

**package** packoops;

**import** java.util.Scanner;

**public** **class** employee {

**int** eNo;

String eName;

**int** salary;

employee(**int** a, String b,**int** c) {

eNo=a;

eName=b;

salary=c;

}

**public** **void** display(**int** p)

{

**if**(p==eNo) {

System.***out***.println("Employee number : " + eNo +"\n"+ "Employee Name : " +eName + "\n"+ "Salary : " +salary);

}

}

**public** **static** **void** main(String[] args)

{ **int** a;

**int** b;

**int** s;

String c;

Scanner obq=**new** Scanner(System.***in***);

Scanner obq1=**new** Scanner(System.***in***);

System.***out***.println("Enter the number of employee :");

a=obq.nextInt();

employee[] obj=**new** employee[a];

**for**(**int** i=0;i<a;i++)

{

System.***out***.println("Enter the employee N.O :");

b=obq.nextInt();

System.***out***.println("Enter the employee Name :");

c=obq1.nextLine();

System.***out***.println("Enter the employee Salary :");

s=obq.nextInt();

obj[i]=**new** employee(b,c,s);

}

System.***out***.println("Enter the emp number :");

**int** t=obq.nextInt();

**for**(**int** j=0;j<a;j++)

{

obj[j].display(t);

}

}

}

**Output:**

