OBJECT ORIENTED PROGRAMMING LAB

Submitted By:

Nandana Anil

S2 MCA

Roll No:224

LAB CYCLE 1

PROGRAM 1

AIM

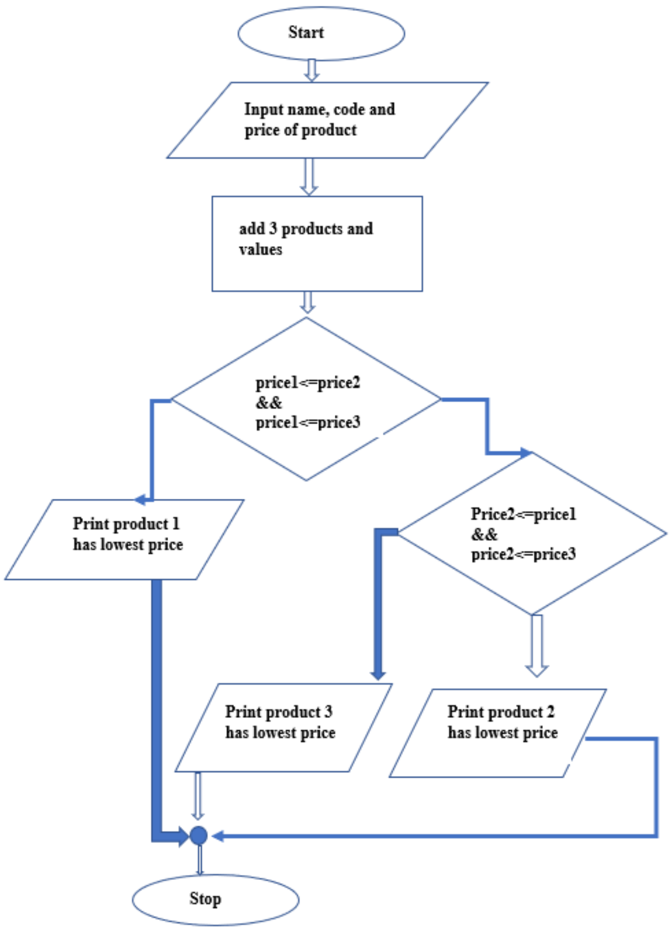
Define a class ‘product’ with data members pcode, pname, and price. Create three objects of the class and find the product having the lowest price.

ALGORITHM

STEP 1: Start
STEP 2: Define a class name as product with members pname, pcode and price.
STEP 3: Define objects to Class and add 3 products and values to each data using the object.
STEP 4: Check whether which product has lowest price using if-else statement.
STEP 5: Print the details of product.
STEP 6: Stop

FLOWCHART

|


Start
Input name, code and price of product
add 3 products and values
price1<=price2 && price1<=price3
Price2<=price1 && price2<=price3
Print product 1 has lowest price
Print product 3 has lowest price
Print product 2 has lowest price
Stop

PROGRAM CODE	see code below

PROGRAM CODE

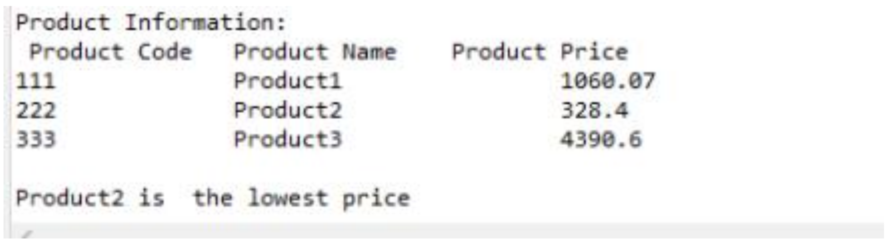
```
public class product{
    int pcode;
    String pname;
    double price;
    double lowest;
    void data(int c, String n, double p){
        pcode=c;
        pname=n;
        price=p;
    }
    void display(){

  System.out.println(pcode+"\t\t"+pname+"\t\t"+price);

    }
    static void findLowest(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){
        product obj1 = new product();
        product obj2 = new product();
        product obj3 = new product();
        obj1.data(111,"Product1",1060.07);
        obj2.data(222,"Product2",328.40);
        obj3.data(333,"Product3",4390.60);
        System.out.println("Product Information:\n
  Product Code\tProduct Name\tProduct Price");
```

	```
	        obj1.display();
	        obj2.display();
	        obj3.display();
	        findLowest(obj1.price,obj2.price,obj3.price);

	      }
	    }
	```

RESULT

The above program is executed and obtained the output

OUTPUT



```
Product Information:
 Product Code   Product Name    Product Price
111             Product1                1060.07
222             Product2                328.4
333             Product3                4390.6

Product2 is  the lowest price
```

PROGRAM 2

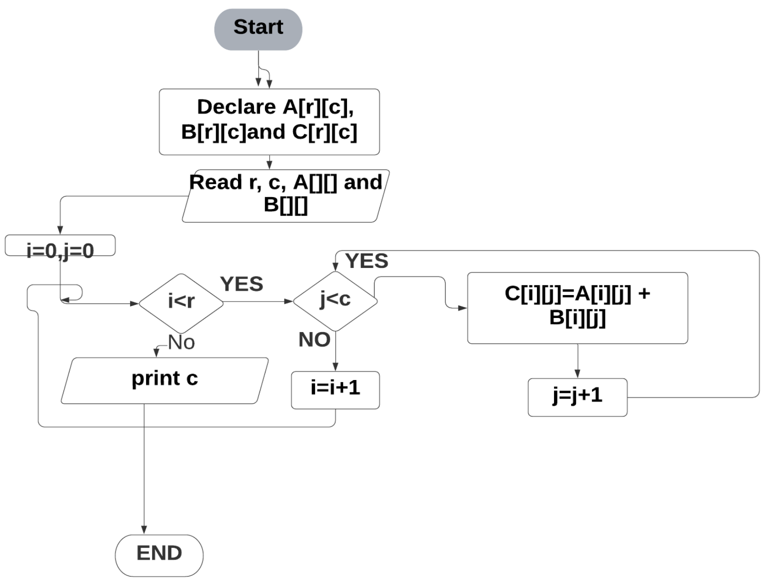
AIM

Read two matrices from the console and perform matrix addition.

ALGORITHM

```
STEP 1: Start
STEP 2: Declare matrix A[r][c];and matrix B[r][c];and matrix C[r][c]; r= no. of rows, c= no. of columns
STEP 3: Read r, c, A[][] and B[][]
STEP 4: Declare variable i=0, j=0
STEP 5: Repeat until i < r
        5.1: Repeat until j < c
             C[i][j] = A[i][j] + B[i][j]
             Set j=j+1
        5.2: Set i=i+1
STEP 6: C is the required matrix after addition
STEP 7: Stop
```

FLOWCHART


Start
Declare A[r][c], B[r][c]and C[r][c]
Read r, c, A[][] and B[][]
i=0,j=0
i<r
YES
No
print c
j<c
YES
NO
i=i+1
C[i][j]=A[i][j] + B[i][j]
j=j+1
END

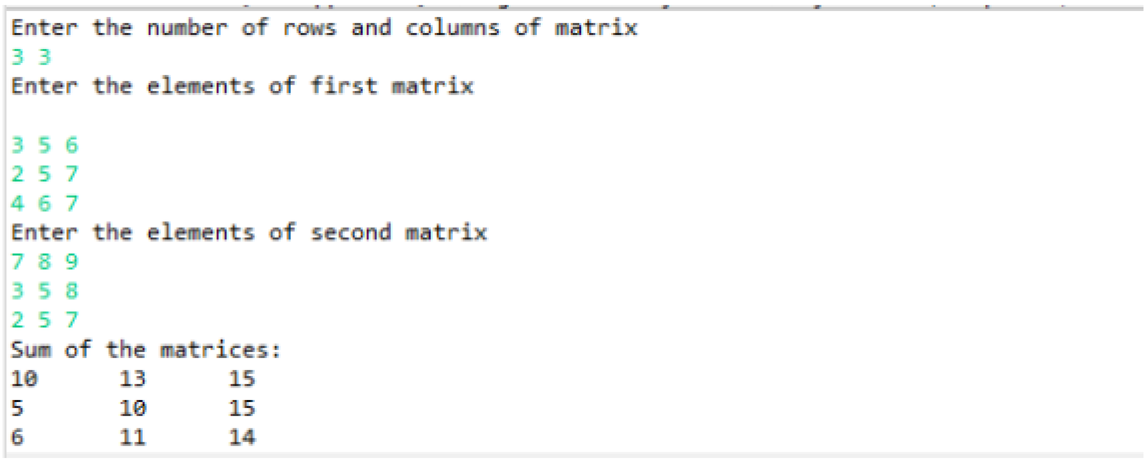
PROGRAM CODE

```
import java.util.Scanner;
class console
{
  public static void main(String args[])
  {
    int m, n, c, d;
    Scanner in = new Scanner(System.in);

    System.out.println("Enter the number of
rows and columns of matrix");
    m = in.nextInt();
    n = in.nextInt();

    int first[][] = new int[m][n];
    int second[][] = new int[m][n];
    int sum[][] = new int[m][n];

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

    for (c = 0; c < m; c++)
      for (d = 0; d < n; d++)
        first[c][d] = in.nextInt();

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

    for (c = 0 ; c < m; c++)
      for (d = 0 ; d < n; d++)
        second[c][d] = in.nextInt();

    for (c = 0; c < m; c++)
      for (d = 0; d < n; d++)
        sum[c][d] = first[c][d] + second[c][d];
//replace '+' with '-' to subtract matrices

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

    for (c = 0; c < m; c++)
    {
      for (d = 0; d < n; d++)
        System.out.print(sum[c][d] + "\t");

      System.out.println();
    }
  }
}
```

RESULT

The above program is successfully executed and obtained the output.

OUTPUT



```
Enter the number of rows and columns of matrix
3 3
Enter the elements of first matrix

3 5 6
2 5 7
4 6 7
Enter the elements of second matrix
7 8 9
3 5 8
2 5 7
Sum of the matrices:
10      13      15
5       10      15
6       11      14
```

PROGRAM 3

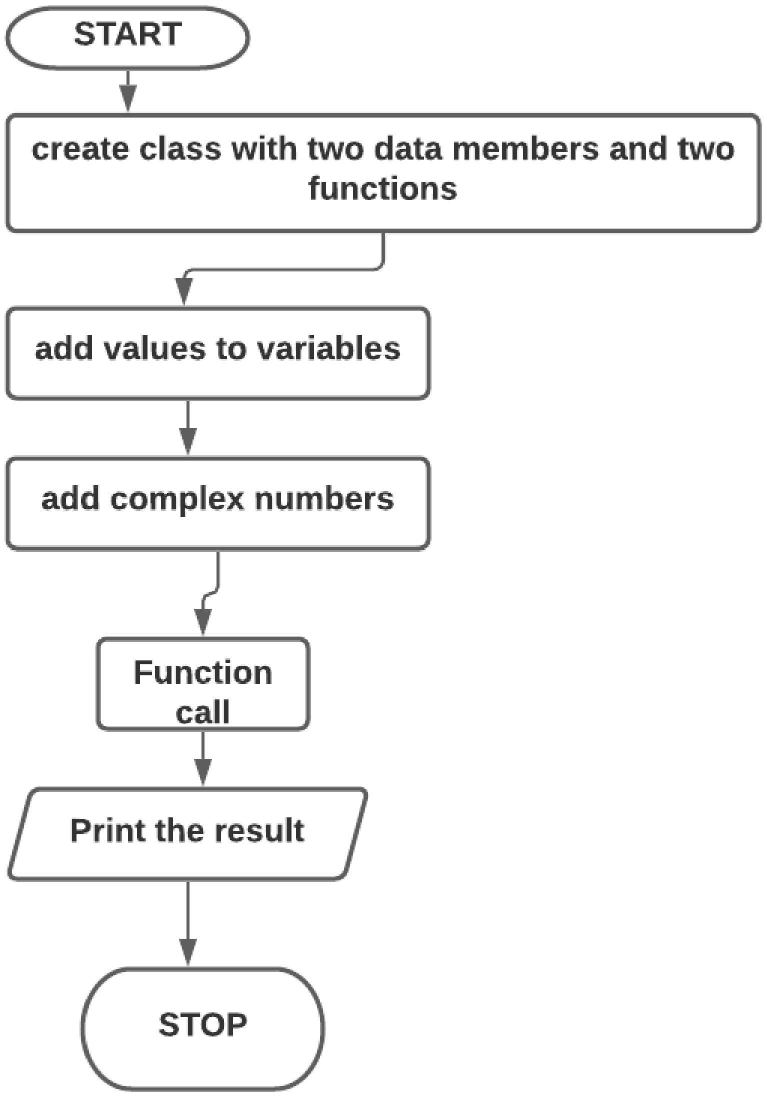
AIM

Add complex numbers.

ALGORITHM

```
STEP 1: Start
STEP 2: Create class with 2 data members and 2 functions.
STEP 3: First function is used to add values to variables.
STEP 4: Second function is used to add the complex numbers and return the value.
STEP 5: Define object to call the function and Print the result.
STEP 6: Stop
```

FLOWCHART

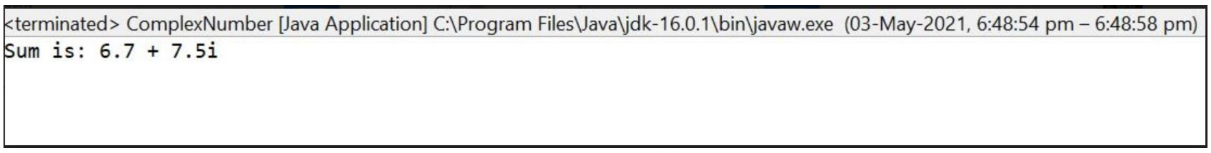


PROGRAM CODE	
	```
	public class ComplexNumber {
	        double real, img;
	ComplexNumber(double r, double i){
	        this.real = r;
	        this.img = i;
	}

	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[]) {
	        ComplexNumber c1 = new ComplexNumber(5.5, 4);
	        ComplexNumber c2 = new ComplexNumber(1.2, 3.5);
	    ComplexNumber temp = sum(c1, c2);
	    System.out.printf("Sum is: "+ temp.real+" + "+ temp.img +"i");
	 }
	}
	```

RESULT

The above program is successfully executed and obtained the output.

OUTPUT



```
<terminated> ComplexNumber [Java Application] C:\Program Files\Java\jdk-16.0.1\bin\javaw.exe  (03-May-2021, 6:48:54 pm – 6:48:58 pm)
Sum is: 6.7 + 7.5i
```

PROGRAM 4

AIM

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

ALGORITHM

STEP 1: Start

STEP 2: Read a matrix using for loop.

STEP 3: Check the number of rows and columns are same. If its same;

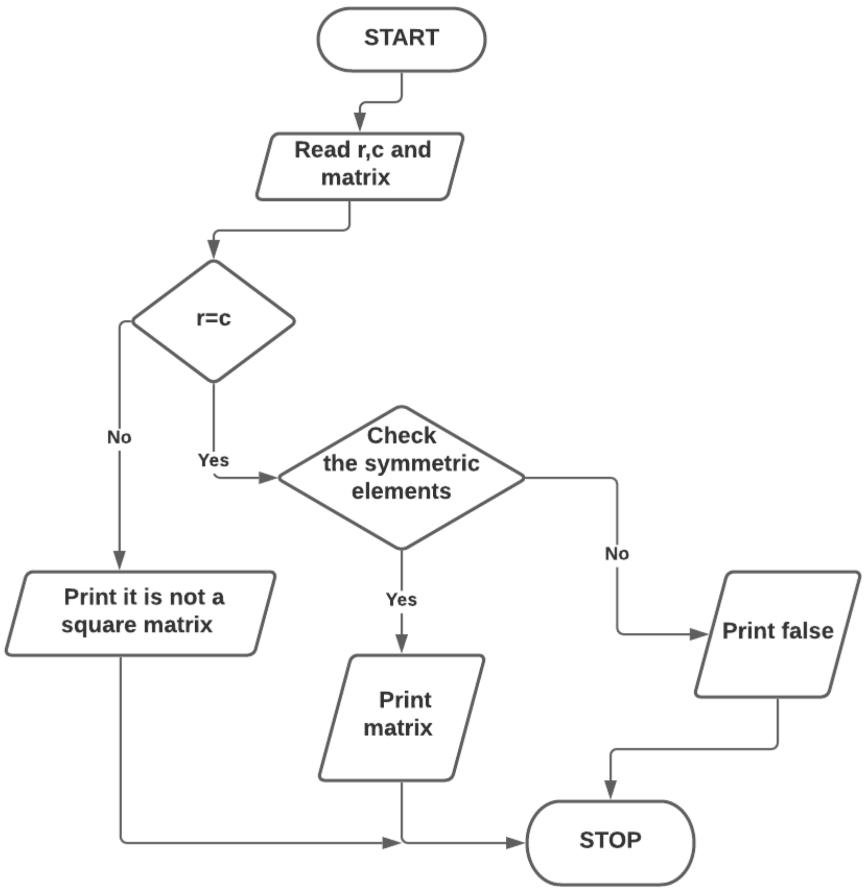
STEP 4: Check the symmetric elements are same. If its same;

STEP 5: Print the matrix and Print its True.

STEP 6: Else print its false.

STEP 7: Stop

FLOWCHART



PROGRAM CODE

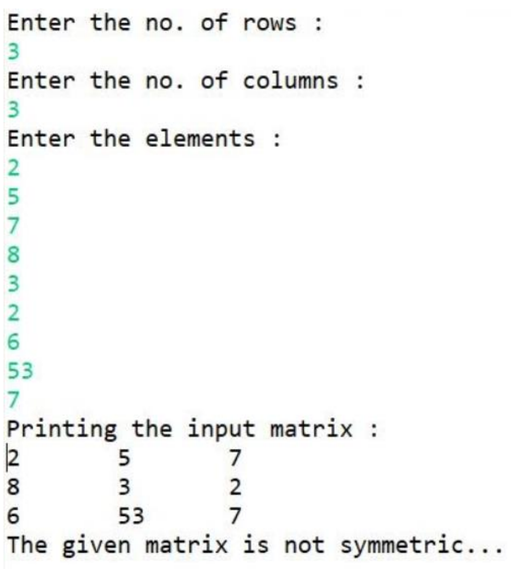
```
import java.util.Scanner;
public class SymmetricMatrixProgram
{public static void main(String[] args)
{
        Scanner sc = new Scanner(System.in);

        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();
}

}
```

RESULT

The above program is successfully executed and obtained the output.

OUTPUT



```
Enter the no. of rows :
3
Enter the no. of columns :
3
Enter the elements :
2
5
7
8
3
2
6
53
7
Printing the input matrix :
2       5       7
8       3       2
6       53      7
The given matrix is not symmetric...
```

PROGRAM 5

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 members as price and a class processor.

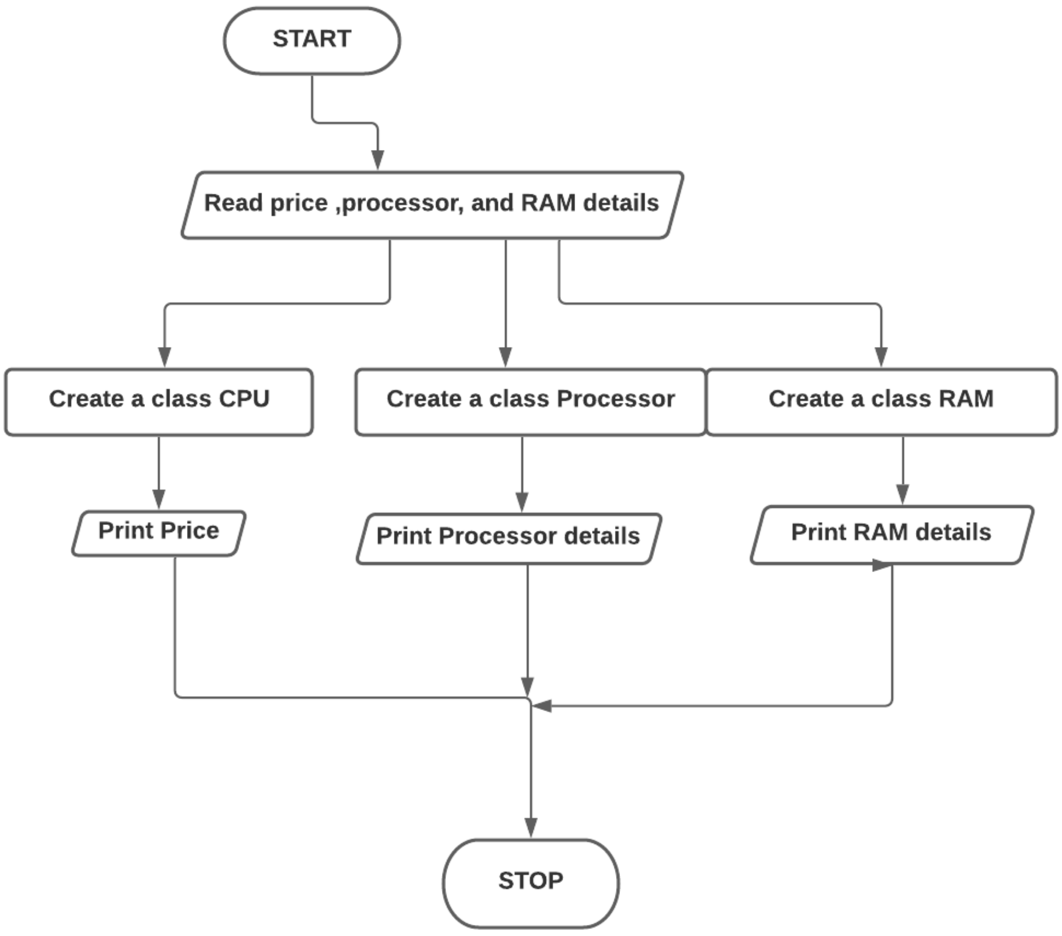
STEP 3: Class processor contain members as cores, manufacture and nested class Ram.

STEP 5: Class Ram contain members as memory and, manufactures.

STEP 6: Create objects for each class and Print its details.

STEP 7: Stop

FLOWCHART



<u>PROGRAM CODE</u>

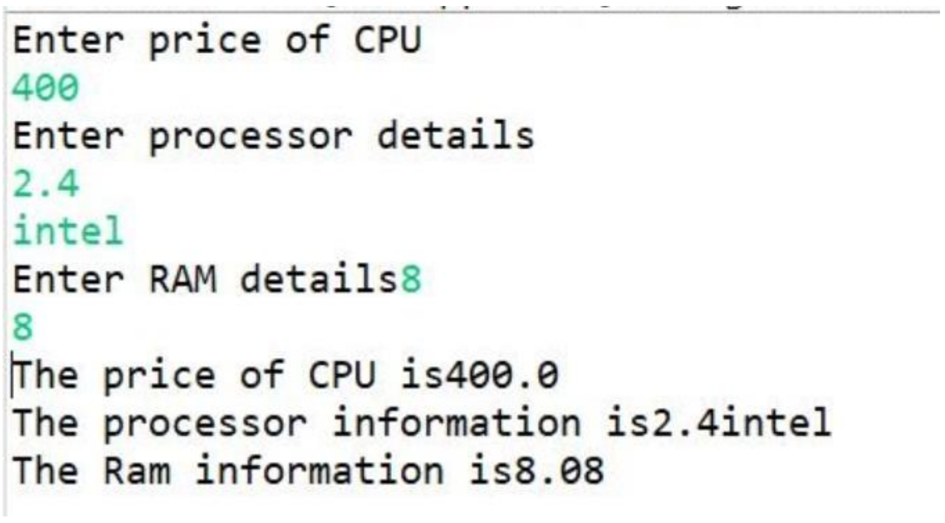
```
package JAVA;
import java.util.Scanner;
import java.lang.String;

public class CPU {
          double price;
            public class processor{
                      float ncores;
                      String manufacturer;
                      void pinfo(float
a,String processorname) {
                           ncores=a;

manufacturer=processorname;

System.out.println("The processor
information is" +ncores+ "" +manufacturer);
            }
            }
            static class ram{
                      float memory;
                      String manufacturer;
                      void prinfo(float
b,String ramname) {
                                memory=b;

manufacturer=ramname;


System.out.println("The Ram information
is" +memory+ "" +manufacturer);

            }
            }

                    public static void
main(String[] args) {
                              CPU obj=new
CPU();
                              CPU.processor
obj1=obj.new processor();
                              CPU.ram
obj2=new CPU.ram();
                              Scanner
sc=new Scanner(System.in);

          System.out.println("Enter price of
CPU");

          obj.price=sc.nextInt();
```

	System.out.println("Enter processor details"); float a=sc.nextFloat(); Scanner sc1=new Scanner(System.in); String processorname=sc1.nextLine(); System.out.print("Enter RAM details"); float b=sc.nextFloat(); String ramname=sc1.nextLine(); sc.close(); sc1.close(); System.out.println("The price of CPU is"+obj.price); obj1.pinfo(a, processorname); obj2.prinfo(b, ramname); } }

RESULT

The above program is successfully executed and obtained the output.

OUTPUT



```
Enter price of CPU
400
Enter processor details
2.4
intel
Enter RAM details8
8
The price of CPU is400.0
The processor information is2.4intel
The Ram information is8.08
```