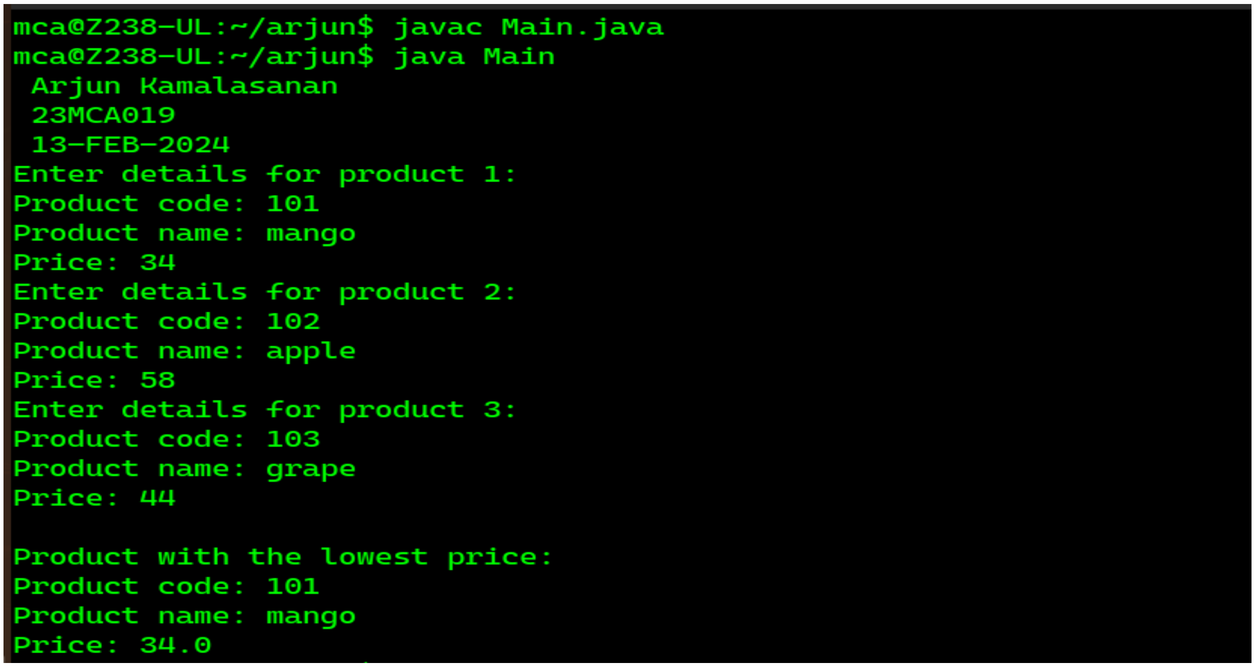
1. Define a class 'product' with data members pcode, pname and price. Create 3 objects of the class and find the product having the lowest price.

Program:

```
import java.util.Scanner;

class Product {
    String pcode;
    String pname;
    double price;

    public Product(String pcode, String pname, double price) {
        this.pcode = pcode;
        this.pname = pname;
        this.price = price;
    }
}

class Main {
    public static void main(String[] args) {

        System.out.println("Arjun Kamalasanan");
        System.out.println("23mca019");
        System.out.println("13-02-24");
        System.out.println("\n");

        Scanner scanner = new Scanner(System.in);

        Product[] products = new Product[3];

        for (int i = 0; i < products.length; i++) {
            System.out.println("Enter details for product " + (i + 1) + ":");
            System.out.print("Product code: ");
            String pcode = scanner.nextLine();
            System.out.print("Product name: ");
            String pname = scanner.nextLine();
            System.out.print("Price: ");
            double price = scanner.nextDouble();
            scanner.nextLine();
            products[i] = new Product(pcode, pname, price);

        }

        Product lowestPriceProduct = products[0];
        for (int i = 1; i < products.length; i++) {
            if (products[i].price < lowestPriceProduct.price) {
                lowestPriceProduct = products[i];
            }
        }


        System.out.println("\nProduct with the lowest price:");
        System.out.println("Product code: " + lowestPriceProduct.pcode);
        System.out.println("Product name: " + lowestPriceProduct.pname);
        System.out.println("Price: " + lowestPriceProduct.price);

        scanner.close();
    }
}
```

Output:



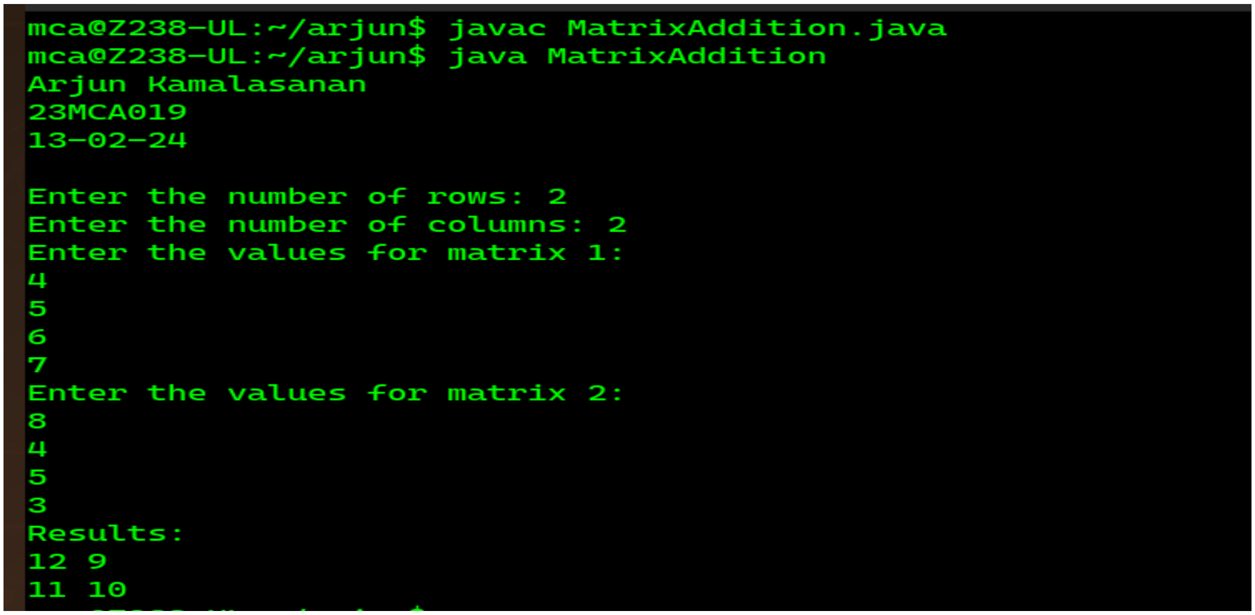
```
mca@Z238-UL:~/arjun$ javac Main.java
mca@Z238-UL:~/arjun$ java Main
 Arjun Kamalasanan
 23MCA019
 13-FEB-2024
Enter details for product 1:
Product code: 101
Product name: mango
Price: 34
Enter details for product 2:
Product code: 102
Product name: apple
Price: 58
Enter details for product 3:
Product code: 103
Product name: grape
Price: 44

Product with the lowest price:
Product code: 101
Product name: mango
Price: 34.0
```

2. Read 2 matrices from the console and perform matrix addition.

Program:

```
import java.util.Scanner;

        public class MatrixAddition {

        public static void main(String[] args) {

        System.out.println("Arjun Kamalasanan \n23MCA019 \n13-02-24\n");
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of rows: ");
        int rows = scanner.nextInt();
        System.out.print("Enter the number of columns: ");
        int cols = scanner.nextInt();

        int[][] matrix1 = new int[rows][cols];
        System.out.println("Enter the values for matrix 1:");
        for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
        matrix1[i][j] = scanner.nextInt();
        }
        }

        int[][] matrix2 = new int[rows][cols];
        System.out.println("Enter the values for matrix 2:");
        for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
        matrix2[i][j] = scanner.nextInt();
        }
        }

        int[][] result = new int[rows][cols];
        for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
        result[i][j] = matrix1[i][j] + matrix2[i][j];
        }
        }

        System.out.println("Results:");
        for (int i = 0; i < rows; i++) {
        for (int j = 0; j < rows; j++) {
        System.out.print(result[i][j] + " ");
        }
        System.out.println();
        }
scanner.close();
}
}
```

Output:



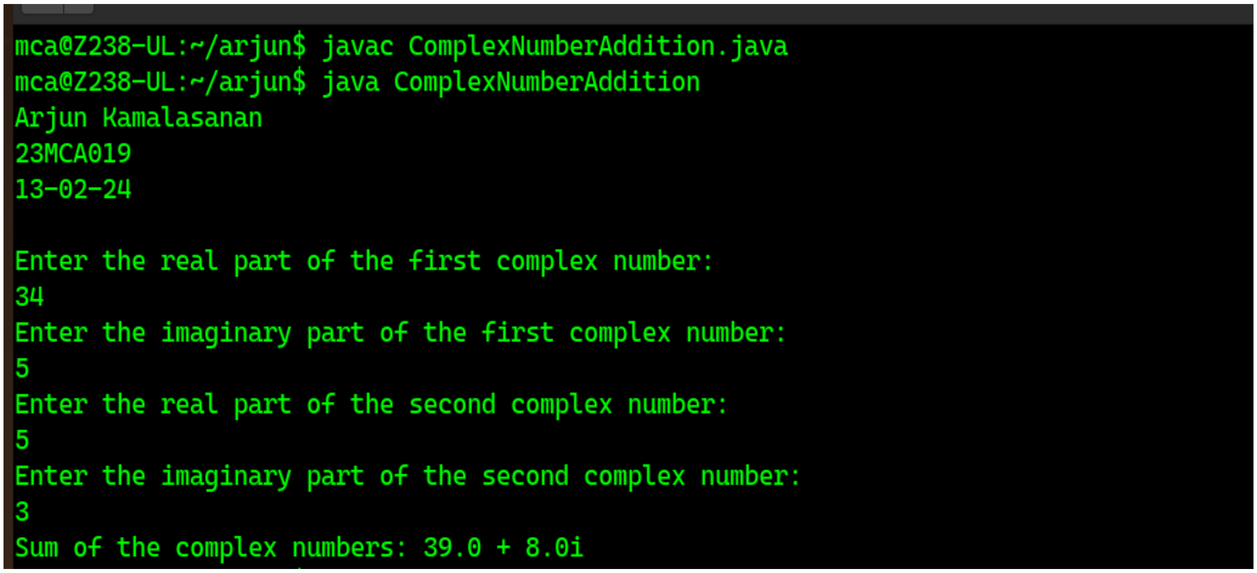
```
mca@Z238-UL:~/arjun$ javac MatrixAddition.java
mca@Z238-UL:~/arjun$ java MatrixAddition
Arjun Kamalasanan
23MCA019
13-02-24

Enter the number of rows: 2
Enter the number of columns: 2
Enter the values for matrix 1:
4
5
6
7
Enter the values for matrix 2:
8
4
5
3
Results:
12 9
11 10
```

3. Add complex numbers

Program:

```
import java.util.Scanner;


    public class ComplexNumberAddition {
        public static void main(String[] args) {

                System.out.println("Arjun Kamalasanan\n23MCA019 \n13-02-24\n");

            Scanner scanner = new Scanner(System.in);


            System.out.println("Enter the real part of the first complex number:");
            double real1 = scanner.nextDouble();
            System.out.println("Enter the imaginary part of the first complex number:");
            double imaginary1 = scanner.nextDouble();


            System.out.println("Enter the real part of the second complex number:");
            double real2 = scanner.nextDouble();
            System.out.println("Enter the imaginary part of the second complex number:");
            double imaginary2 = scanner.nextDouble();


            double sumReal = real1 + real2;
            double sumImaginary = imaginary1 + imaginary2;


            System.out.println("Sum of the complex numbers: " + sumReal + " + " + sumImaginary +
    "i");

            scanner.close();
        }
    }
```

Output:



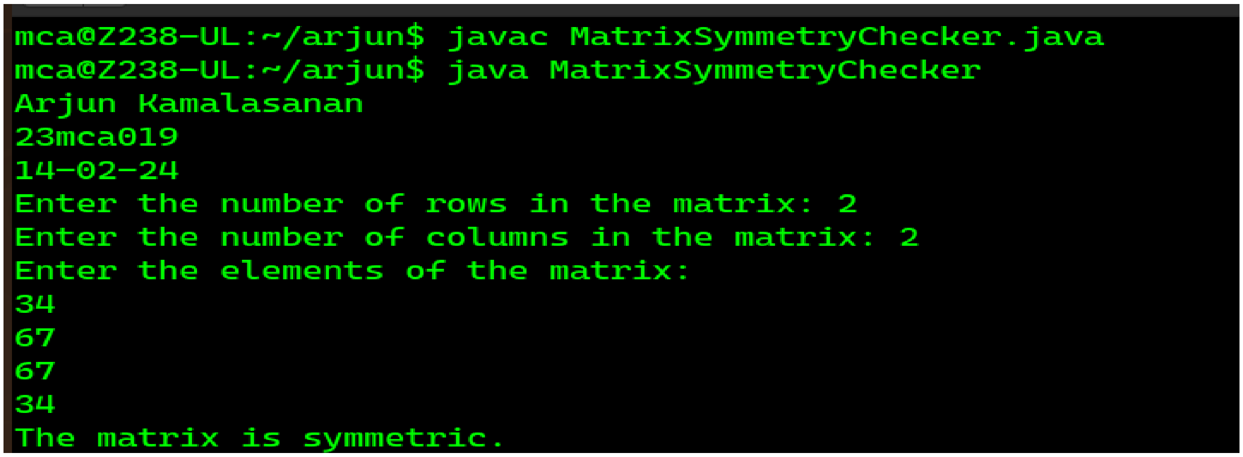
```
mca@Z238-UL:~/arjun$ javac ComplexNumberAddition.java
mca@Z238-UL:~/arjun$ java ComplexNumberAddition
Arjun Kamalasanan
23MCA019
13-02-24

Enter the real part of the first complex number:
34
Enter the imaginary part of the first complex number:
5
Enter the real part of the second complex number:
5
Enter the imaginary part of the second complex number:
3
Sum of the complex numbers: 39.0 + 8.0i
```

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

Program:

```
import java.util.Scanner;

        public class MatrixSymmetryChecker {

        public static void main(String[] args) {

        System.out.println("Arjun Kamalasanan \n23mca019 \n14-02-24");
        Scanner input = new Scanner(System.in);
        System.out.print("Enter the number of rows in the matrix: ");

        int rows = input.nextInt();
        System.out.print("Enter the number of columns in the matrix: ");
        int cols = input.nextInt();
        int[][] matrix = new int[rows][cols];
        System.out.println("Enter the elements of the matrix:");
        for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
        matrix[i][j] = input.nextInt();
        }
        }
        boolean isSymmetric = true;
        for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
        if (matrix[i][j] != matrix[j][i]) {
        isSymmetric = false;
        break;
        }
        }
        if (!isSymmetric) {
        break;
        }
        }
        if (isSymmetric) {
        System.out.println("The matrix is symmetric.");
        } else {
        System.out.println("The matrix is not symmetric.");
        }
        }
        }
```

Output:



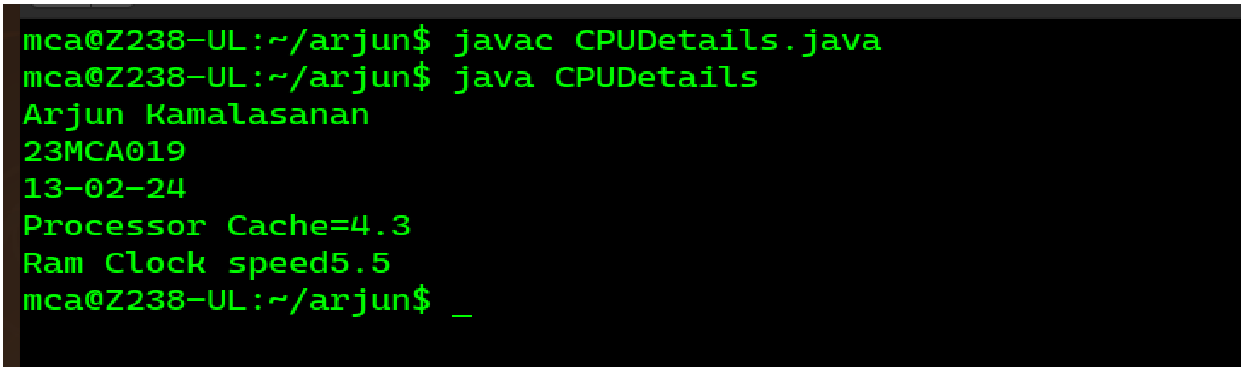
```
mca@Z238-UL:~/arjun$ javac MatrixSymmetryChecker.java
mca@Z238-UL:~/arjun$ java MatrixSymmetryChecker
Arjun Kamalasanan
23mca019
14-02-24
Enter the number of rows in the matrix: 2
Enter the number of columns in the matrix: 2
Enter the elements of the matrix:
34
67
67
34
The matrix is symmetric.
```

5. 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.

Program:

```
class CPU
{
        double price;
        class Processor
        {
                double cores;
                String manufacturer;
                double getCache()
                {
                        return 4.3;
                }
        }
        protected class RAM
        {
                double memory;
                String manufacturer;
                double getClockSpeed()
                {
                        return 5.5;
                }
        }
}
public class CPUDetails
{
        public static void main(String[] args)
        {
        System.out.println("Arjun Kamalasanan \n23MCA019\n13-02-24");
        CPU cpu=new CPU();
        CPU.Processor processor=cpu.new Processor();
        CPU.RAM ram=cpu.new RAM();
        System.out.println("Processor Cache=" + processor.getCache());
        System.out.println("Ram Clock speed" + ram.getClockSpeed());
        }
}
```

Output:



```
mca@Z238-UL:~/arjun$ javac CPUDetails.java
mca@Z238-UL:~/arjun$ java CPUDetails
Arjun Kamalasanan
23MCA019
13-02-24
Processor Cache=4.3
Ram Clock speed5.5
mca@Z238-UL:~/arjun$ _
```