

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.

Source code:

```
public class Product {  
    String pname, pcode;  
    int price;  
  
    public String getPname() {  
        return pname;  
    }  
  
    public Product() {  
    }  
  
    public Product(String pname, String pcode, int price) {  
        this.pname = pname;  
        this.pcode = pcode;  
        this.price = price;  
    }  
  
    public void setName(String pname) {  
        this.pname = pname;  
    }  
  
    public String getPcode() {  
        return pcode;  
    }  
  
    public void setPcode(String pcode) {  
        this.pcode = pcode;  
    }  
  
    public int getPrice() {
```

```
return price;
}

public void setPrice(int price) {
    this.price = price;
}

public void display() {
    System.out.println("pcode: " + this.pcode);
    System.out.println("pname: " + this.pname);
    System.out.println("price: " + this.price);
}

public static void main(String[] args) {
    System.out.println("Name: Shijo jose");
    System.out.println("Roll.no: 51");
    System.out.println("Title: Define a class 'product' with data members pcode, pname, and price.");
    System.out.println("Create 3 objects of the class and find the product having the lowest price.");

    Product p1 = new Product();
    p1.setPcode("Car123");
    p1.setPname("Benz");
    p1.setPrice(100000);
    System.out.println("\n*****Displaying p1*****");
    p1.display();

    Product p2 = new Product("Jaguar", "Car246", 25000);
    System.out.println("\n*****Displaying p2*****");
    p2.display();

    Product p3 = new Product("Maruthi", "Car800", 50000);
    System.out.println("\n*****Displaying p3*****");
    p3.display();
}
```

```
Product productWithLowestPrice = (p3.getPrice() < Math.min(p1.getPrice(), p2.getPrice())) ?  
p3 : (p1.getPrice() < p2.getPrice() ? p1 : p2);  
System.out.println("\n*****Displaying product with the lowest price*****");  
productWithLowestPrice.display();  
}  
}
```

Output:

```
mca@HP-Z238:~/Downloads$ javac Product.java  
mca@HP-Z238:~/Downloads$ java Product  
Name: Shijo jose  
Roll.no: 51  
Title: 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.  
  
*****Displaying p1*****  
pcode: Car123  
pname: Benz  
price: 100000  
  
*****Displaying p2*****  
pcode: Car246  
pname: Jaguar  
price: 25000  
  
*****Displaying p3*****  
pcode: Car800  
pname: Maruthi  
price: 50000  
  
*****Displaying product with the lowest price*****  
pcode: Car246  
pname: Jaguar  
price: 25000  
mca@HP-Z238:~/Downloads$
```

2. Read 2 matrices from the console and perform matrix addition.**Source code:**

```
import java.util.Scanner;

public class Matrix {
    public static void main(String[] args) {
        System.out.println("name:shijo jose\nRoll no:51");
        Scanner scanner = new Scanner(System.in);

        System.out.println("Enter the number of rows and columns for the first matrix:");
        int row1 = scanner.nextInt();
        int col1 = scanner.nextInt();
        int[][] mat1 = inputMatrix(scanner, row1, col1);

        System.out.println("Enter the number of rows and columns for the second matrix:");
        int row2 = scanner.nextInt();
        int col2 = scanner.nextInt();
        int[][] mat2 = inputMatrix(scanner, row2, col2);

        System.out.println("First matrix:");
        displayMatrix(mat1);
        System.out.println("Second matrix:");
        displayMatrix(mat2);

        if (row1 == row2 && col1 == col2) {
            int[][] sum = addMatrices(mat1, mat2);
            System.out.println("Sum of matrices:");
            displayMatrix(sum);
        } else {
            System.out.println("Addition not possible. Matrices have different dimensions.");
        }

        scanner.close();
    }
}
```

```
}

public static int[][] inputMatrix(Scanner scanner, int rows, int cols) {
    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] = scanner.nextInt();
        }
    }
    return matrix;
}

public static void displayMatrix(int[][] matrix) {
    for (int[] row : matrix) {
        for (int value : row) {
            System.out.print(value + "\t");
        }
        System.out.println();
    }
}

public static int[][] addMatrices(int[][] mat1, int[][] mat2) {
    int rows = mat1.length;
    int cols = mat1[0].length;
    int[][] sum = new int[rows][cols];
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            sum[i][j] = mat1[i][j] + mat2[i][j];
        }
    }
    return sum;
}
```

Output:

```
1 11
mca@HP-Z238:~/Downloads$ javac Matrix.java
mca@HP-Z238:~/Downloads$ java Matrix
name:shijo jose
Roll no:51
Enter the number of rows and columns for the first matrix:
2
2
Enter the elements of the matrix:
3
4
5
6
Enter the number of rows and columns for the second matrix:
2
2
Enter the elements of the matrix:
6
7
8
9
First matrix:
3      4
5      6
Second matrix:
6      7
8      9
Sum of matrices:
9      11
13     15
mca@HP-Z238:~/Downloads$
```

3. Add complex numbers

Source code:

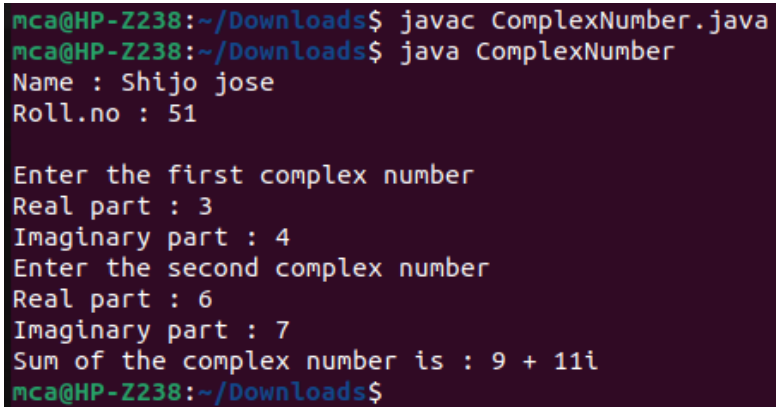
```
import java.util.Scanner;

public class ComplexNumber{
    public static void main(String[] arg){
        System.out.println("Name : Shijo jose\nRoll.no : 51\n");
        class ComplexNumberClass{
            int real;
            int img;
            public ComplexNumberClass(int r, int i) {
                this.real = r;
                this.img = i;
            }
        }
        int a, b;
        Scanner scan = new Scanner(System.in);
        System.out.println("Enter the first complex number");
        System.out.print("Real part : ");
        a=scan.nextInt();
        System.out.print("Imaginary part : ");
        b=scan.nextInt();
        ComplexNumberClass first = new ComplexNumberClass(a, b);

        System.out.println("Enter the second complex number");
        System.out.print("Real part : ");
        a=scan.nextInt();
        System.out.print("Imaginary part : ");
        b=scan.nextInt();
        ComplexNumberClass second = new ComplexNumberClass(a, b);

        int real = first.real + second.real;
        int img = first.img + second.img;
```

```
System.out.println("Sum of the complex number is : " + real + " + " + img + "i");  
}  
}
```

Output:

```
mca@HP-Z238:~/Downloads$ javac ComplexNumber.java  
mca@HP-Z238:~/Downloads$ java ComplexNumber  
Name : Shijo jose  
Roll.no : 51  
  
Enter the first complex number  
Real part : 3  
Imaginary part : 4  
Enter the second complex number  
Real part : 6  
Imaginary part : 7  
Sum of the complex number is : 9 + 11i  
mca@HP-Z238:~/Downloads$
```


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

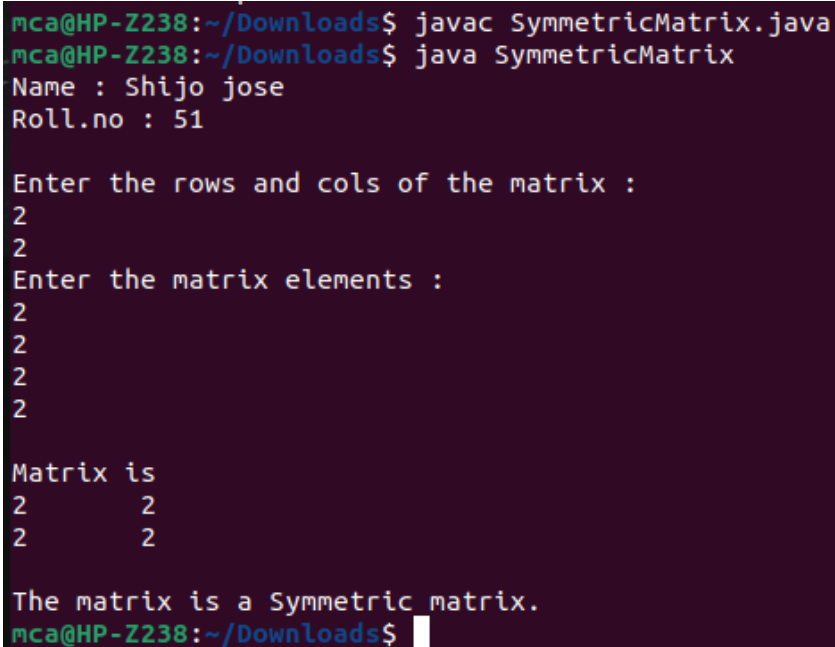
Source code:

```
import java.util.Scanner;

public class SymmetricMatrix{
    public static void main(String[] arg){
        System.out.println("Name : Shijo jose\nRoll.no : 51\n");

        Scanner read = new Scanner(System.in);
        System.out.println("Enter the rows and cols of the matrix : ");
        int rows = read.nextInt();
        int columns = read.nextInt();
        int[][] matrix = new int[rows][columns];
        System.out.println("Enter the matrix elements : ");
        for(int i=0; i<rows; i++){
            for(int j=0; j<columns; j++){
                matrix[i][j] = read.nextInt();
            }
        }
        System.out.println("\nMatrix is");
        for(int i=0; i<rows; i++){
            for(int j=0; j<columns; j++){
                System.out.print(matrix[i][j] + "\t");
            }
            System.out.print("\n");
        }
        if(rows == columns){
            int flag=0;
            for(int i=0; i<rows; i++){
                for(int j=0; j<columns; j++){
                    if(matrix[i][j] != matrix[j][i]){
                        flag=1;
                    }
                }
            }
            if(flag == 1){
                System.out.println("Matrix is not symmetric");
            } else {
                System.out.println("Matrix is symmetric");
            }
        } else {
            System.out.println("Matrix is not square, hence not symmetric");
        }
    }
}
```

```
break;
}
}
if(flag == 1) break;
}
if(flag == 0){
System.out.println("\nThe matrix is a Symmetric matrix.");
} else {
System.out.println("\nThe matrix is not a Symmetric matrix.");
}
} else {
System.out.println("\nThe matrix is not a Symmetric matrix.");
}
}
}
```

Output:

```
mca@HP-Z238:~/Downloads$ javac SymmetricMatrix.java
mca@HP-Z238:~/Downloads$ java SymmetricMatrix
Name : Shijo jose
Roll.no : 51

Enter the rows and cols of the matrix :
2
2
Enter the matrix elements :
2
2
2
2

Matrix is
2      2
2      2

The matrix is a Symmetric matrix.
mca@HP-Z238:~/Downloads$
```

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.

Source code:

```
import java.util.Scanner;

public class Cpu {
    public static void main(String[] arg) {
        System.out.println("Name: Shijo jose\nRoll.no: 51\n");

        class CpuClass {
            int price;

            public CpuClass(int price) {
                this.price = price;
            }

            class Processor {
                int noOfCores;
                String manufacturer;

                public Processor(int noOfCores, String manufacturer) {
                    this.noOfCores = noOfCores;
                    this.manufacturer = manufacturer;
                }

                class Ram {
                    int memory;
                    String manufacturer;

                    public Ram(int memory, String manufacturer) {
                        this.memory = memory;
                    }
                }
            }
        }
    }
}
```

```
this.manufacturer = manufacturer;
}
}
}
}

CpuClass cpu = new CpuClass(1000);
CpuClass.Processor processor = cpu.new Processor(8, "Intel");
CpuClass.Processor.Ram ram = processor.new Ram(16, "Samsung");

System.out.println("Cpu price is: " + cpu.price + "\n");
System.out.println("Processor no. of cores is: " + processor.noOfCores);
System.out.println("Processor manufacturer is: " + processor.manufacturer + "\n");
System.out.println("Ram size is: " + ram.memory);
System.out.println("Ram manufacturer is: " + ram.manufacturer);
}
}
```

Output:

```
mca@HP-Z238:~/Downloads$ javac Cpu.java
mca@HP-Z238:~/Downloads$ java Cpu
Name: Shijo jose
Roll.no: 51

Cpu price is: 1000

Processor no. of cores is: 8
Processor manufacturer is: Intel

Ram size is: 16
Ram manufacturer is: Samsung
mca@HP-Z238:~/Downloads$
```