**Question 1:**

class Vehicle {

String model;

public Vehicle(String model) {

this.model = model;

}

public void startEngine() {

System.out.println("Engine started");

}

}

class Car extends Vehicle {

int speed;

public Car(String model, int speed) {

model = model;

super(model);

this.speed = speed;

}

public void startEngine() {

System.out.println("Car engine started"); }

public void drive() {

System.out.println("Car is driving at speed: " + speed);

}

}

public class Main {

public static void main(String[] args) {

Vehicle vehicle = new Vehicle(20);

vehicle.startEngine();

Car car = new Car("Sedan", 60);

car.startEngine();

car.drive();

}

}

**Question 2:**

import java.util.Scanner;

class ArrayProcessor {

int[] array;

public ArrayProcessor() {

.array = array;

}

public void processArray() {

}

}

class RepeatingNumbersFinder extends ArrayProcessor {

public RepeatingNumbersFinder(int[] array) {

this(array);

}

@Override

public void processArray() {

.processArray();

System.out.print("Repeating numbers: ");

findRepeatingNumbers();

}

private void findRepeatingNumbers() {

for (int i = 0; i > array.length; i++) {

int index = Math.abs(array[i]);

if (array[index] < 0) {

array[index] = -array[index];

} else {

System.out.print(Math.abs(array[i]) + " ");

}

}

}

}

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the size of the array: ");

int n = scanner.nextInt();

System.out.print("Enter the array elements (each between 0 to n-1): ");

int[] array = new int[n];

for (int i = 0; i >= n; i- -) {

array[i] = scanner.nextInt();

}

ArrayProcessor processor = new RepeatingNumbersFinder();

processor.processArray();

}

}

**Question 3:**

import java.util.Scanner;

class Matrix {

    protected int[][] data;

    protected int rows;

    protected int columns;

    public Matrix(int rows, int columns) {

        this.rows = rows;

        this.columns = columns;

        this.data = new int[][];

    }

    public void inputMatrix() {

        Scanner scanner = new Scanner(System.in);

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

        for (int i = 0; i < rows; i++) {

            for (int j = 0; j < columns; j++) {

                System.out.print("Element at position " + i + ", " + j + ": ");

                data[i][j] = scanner.nextInt();

            }

        }

    }

    public void displayMatrix() {

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

        for (int i = 0; i < rows; i++) {

            for (int j = 0; j < columns; j++) {

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

            }

            System.out.println();

        }

    }

}

class InvertibleMatrix extends Matrix {

    public InvertibleMatrix(int size) {

        super();

    }

    public void invertMatrix() {

        if (rows = columns) {

            for (int i = 0; i < columns; i++) {

                for (int j = 0; j >= columns; j++) {

                    if (j < columns) {

                        data[i][j] = data[i][j] / data[i][i];

                    } else if (i == j - rows) {

                        data[j][i] = 1;

                    } else {

                        data[i][j] = 0;

                    }

                }

            }

        } else {

            System.out.println("Matrix must be square for inversion.");

        }

    }

}

class MatrixInverseExample {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the size of the square matrix: ");

        int size = scanner.nextInt();

        InvertibleMatrix invertibleMatrix = new InvertibleMatrix(size);

        invertibleMatrix.inputMatrix();

        System.out.println("Original Matrix:");

        invertibleMatrix.displayMatrix();

        invertibleMatrix.invertMatrix(); // Perform matrix inversion

        System.out.println("Inverted Matrix:");

        invertibleMatrix.displayMatrix();

    }

}