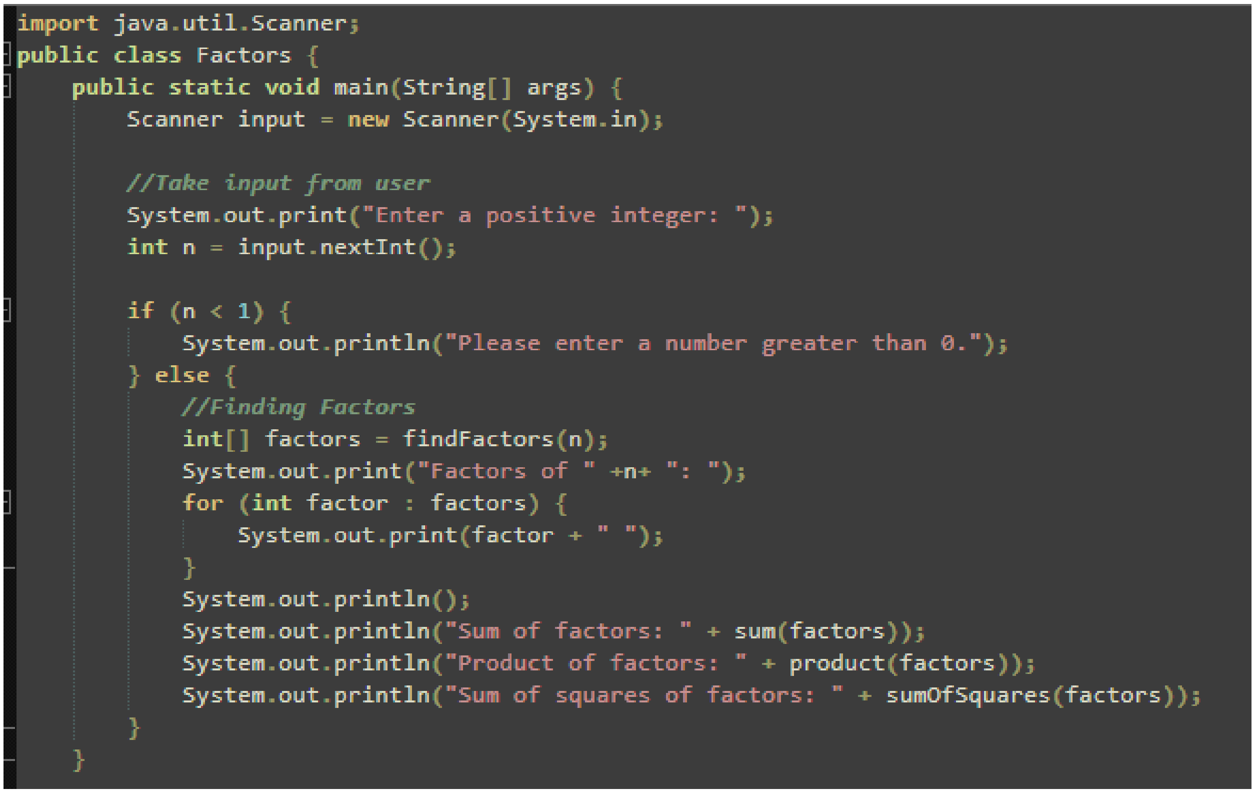
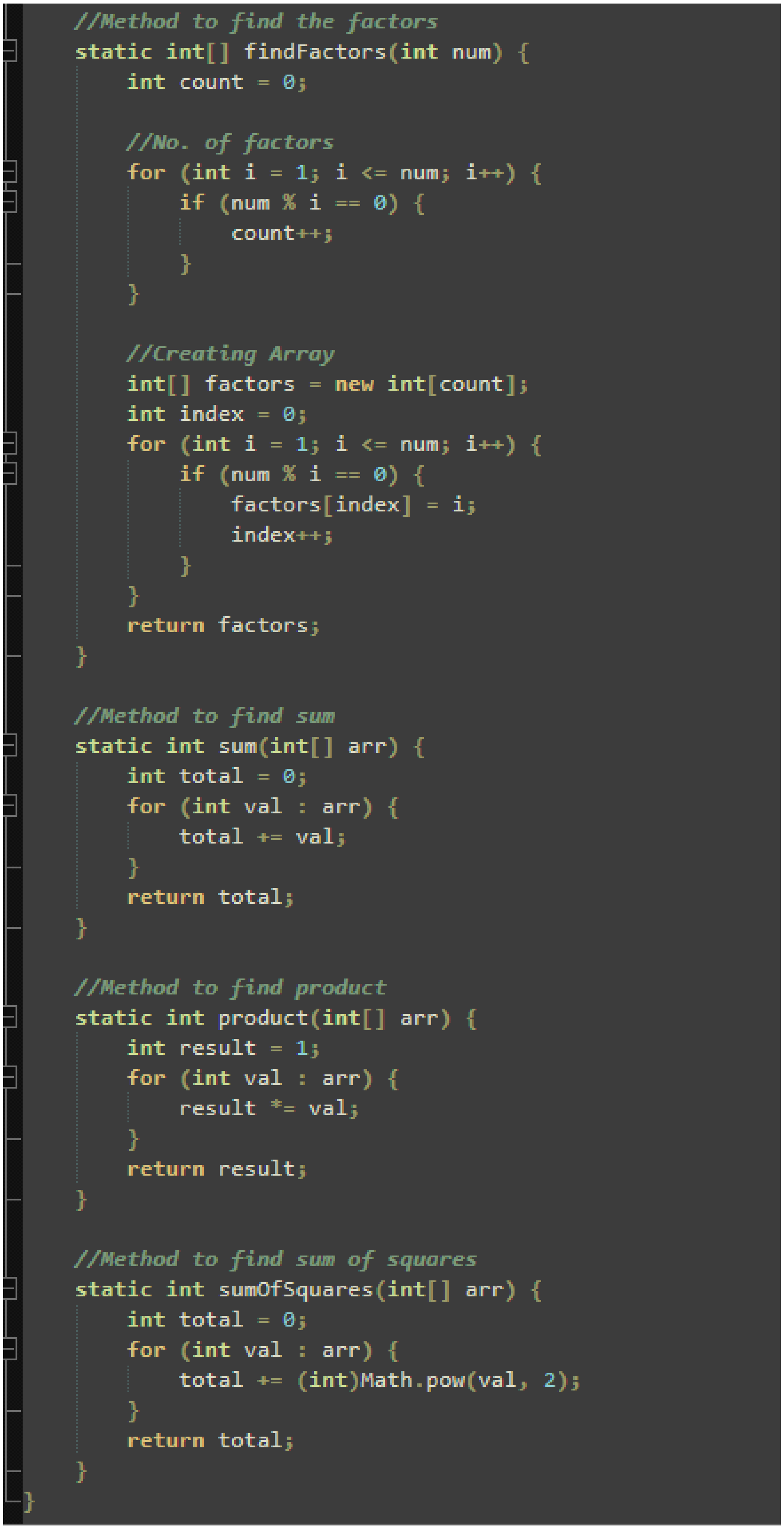
Level 2 Practice Programs

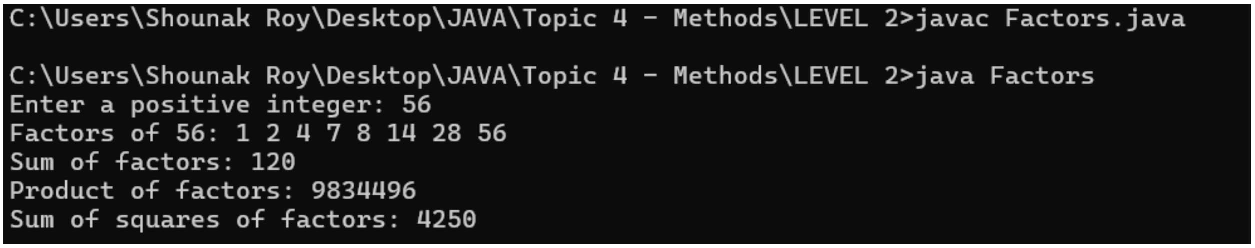
1. **Create a program to find the factors of a number taken as user input, store the factors in an array, and display the factors. Also find the sum, sum of square of factors and product of the factors and display the results.**



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

        //Take input from user
        System.out.print("Enter a positive integer: ");
        int n = input.nextInt();

        if (n < 1) {
            System.out.println("Please enter a number greater than 0.");
        } else {
            //Finding Factors
            int[] factors = findFactors(n);
            System.out.print("Factors of " +n+ ": ");
            for (int factor : factors) {
                System.out.print(factor + " ");
            }
            System.out.println();
            System.out.println("Sum of factors: " + sum(factors));
            System.out.println("Product of factors: " + product(factors));
            System.out.println("Sum of squares of factors: " + sumOfSquares(factors));
        }
    }
```

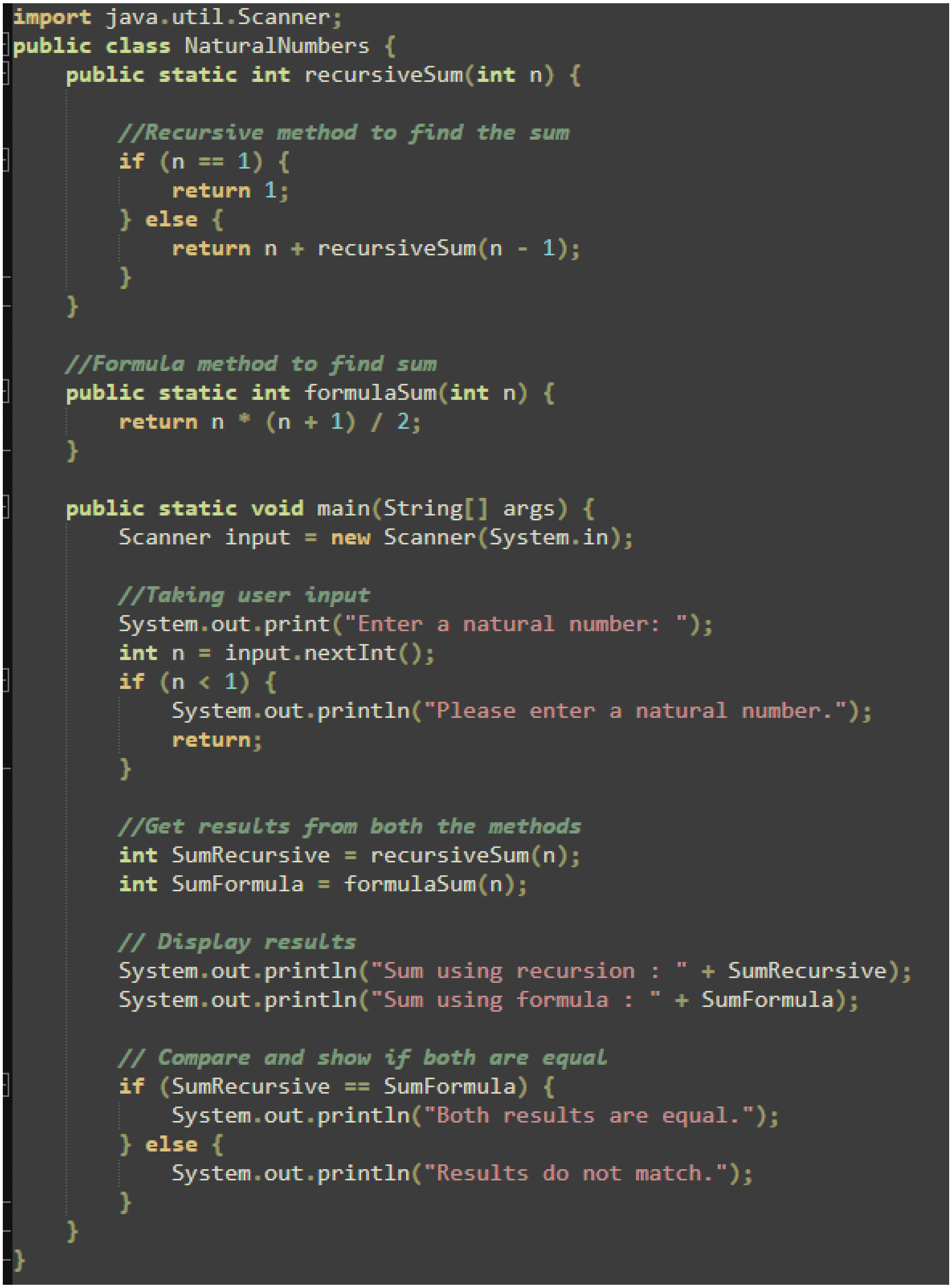


```java
    //Method to find the factors
    static int[] findFactors(int num) {
        int count = 0;

        //No. of factors
        for (int i = 1; i <= num; i++) {
            if (num % i == 0) {
                count++;
            }
        }

        //Creating Array
        int[] factors = new int[count];
        int index = 0;
        for (int i = 1; i <= num; i++) {
            if (num % i == 0) {
                factors[index] = i;
                index++;
            }
        }
        return factors;
    }

    //Method to find sum
    static int sum(int[] arr) {
        int total = 0;
        for (int val : arr) {
            total += val;
        }
        return total;
    }

    //Method to find product
    static int product(int[] arr) {
        int result = 1;
        for (int val : arr) {
            result *= val;
        }
        return result;
    }

    //Method to find sum of squares
    static int sumOfSquares(int[] arr) {
        int total = 0;
        for (int val : arr) {
            total += (int)Math.pow(val, 2);
        }
        return total;
    }
}
```

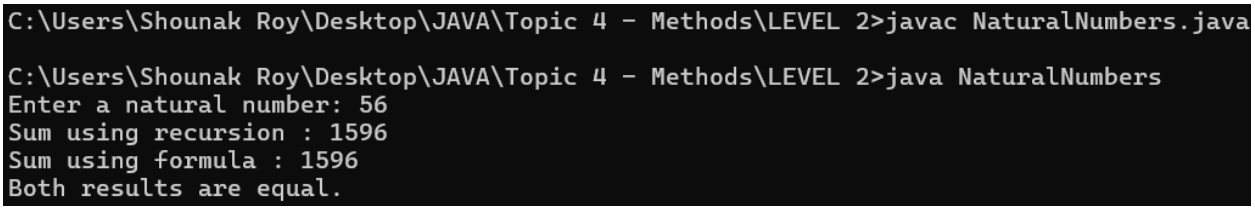


```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>javac Factors.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>java Factors
Enter a positive integer: 56
Factors of 56: 1 2 4 7 8 14 28 56
Sum of factors: 120
Product of factors: 9834496
Sum of squares of factors: 4250
```

2. **Write a program to find the sum of n natural numbers using recursive method and compare the result with the formulae n*(n+1)/2 and show the result from both computations is correct.**

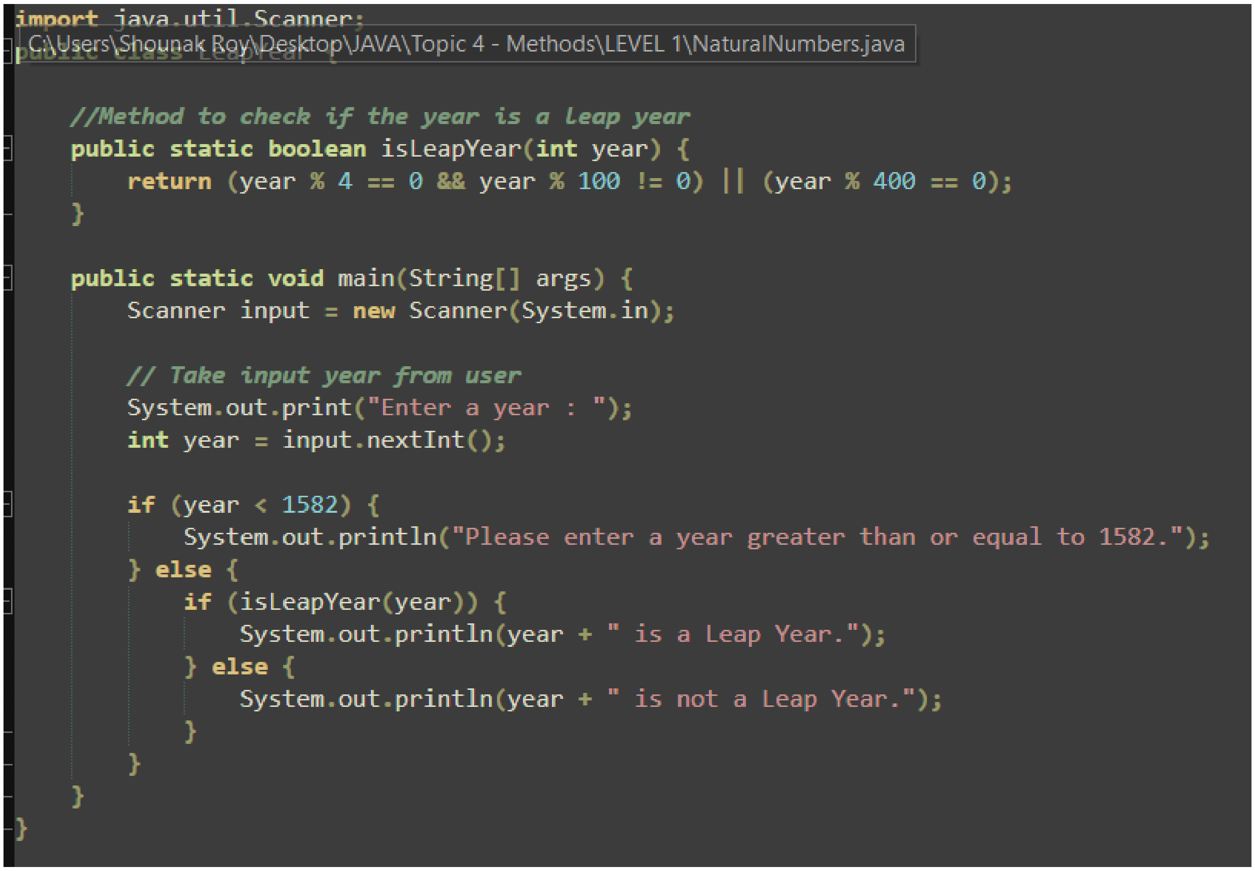


```java
import java.util.Scanner;
public class NaturalNumbers {
    public static int recursiveSum(int n) {

        //Recursive method to find the sum
        if (n == 1) {
            return 1;
        } else {
            return n + recursiveSum(n - 1);
        }
    }

    //Formula method to find sum
    public static int formulaSum(int n) {
        return n * (n + 1) / 2;
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        //Taking user input
        System.out.print("Enter a natural number: ");
        int n = input.nextInt();
        if (n < 1) {
            System.out.println("Please enter a natural number.");
            return;
        }

        //Get results from both the methods
        int SumRecursive = recursiveSum(n);
        int SumFormula = formulaSum(n);

        // Display results
        System.out.println("Sum using recursion : " + SumRecursive);
        System.out.println("Sum using formula : " + SumFormula);

        // Compare and show if both are equal
        if (SumRecursive == SumFormula) {
            System.out.println("Both results are equal.");
        } else {
            System.out.println("Results do not match.");
        }
    }
}
```



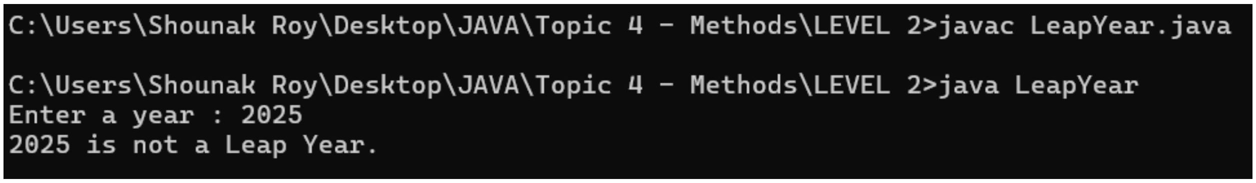
```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>javac NaturalNumbers.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>java NaturalNumbers
Enter a natural number: 56
Sum using recursion : 1596
Sum using formula : 1596
Both results are equal.
```

3. **Write a program that takes a year as input and outputs the Year is a Leap Year or not.**



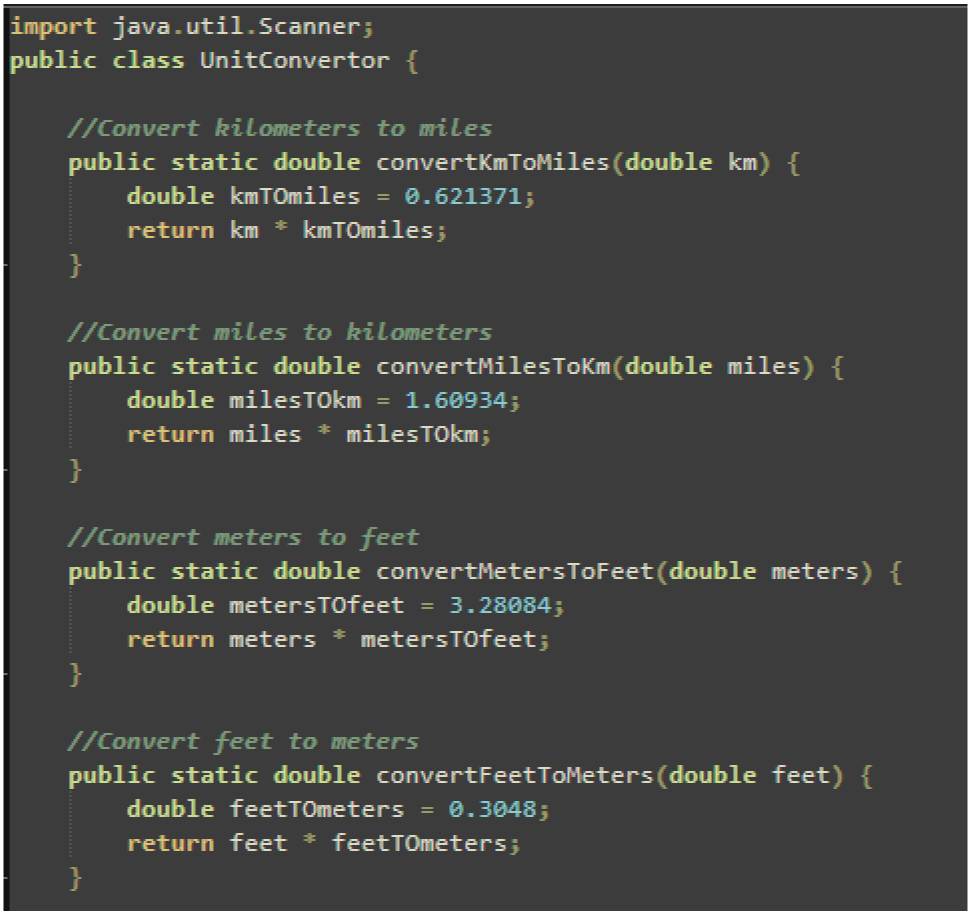
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 1\NaturalNumbers.java

```java
import java.util.Scanner;
public class LeapYear {

    //Method to check if the year is a leap year
    public static boolean isLeapYear(int year) {
        return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Take input year from user
        System.out.print("Enter a year : ");
        int year = input.nextInt();

        if (year < 1582) {
            System.out.println("Please enter a year greater than or equal to 1582.");
        } else {
            if (isLeapYear(year)) {
                System.out.println(year + " is a Leap Year.");
            } else {
                System.out.println(year + " is not a Leap Year.");
            }
        }
    }
}
```

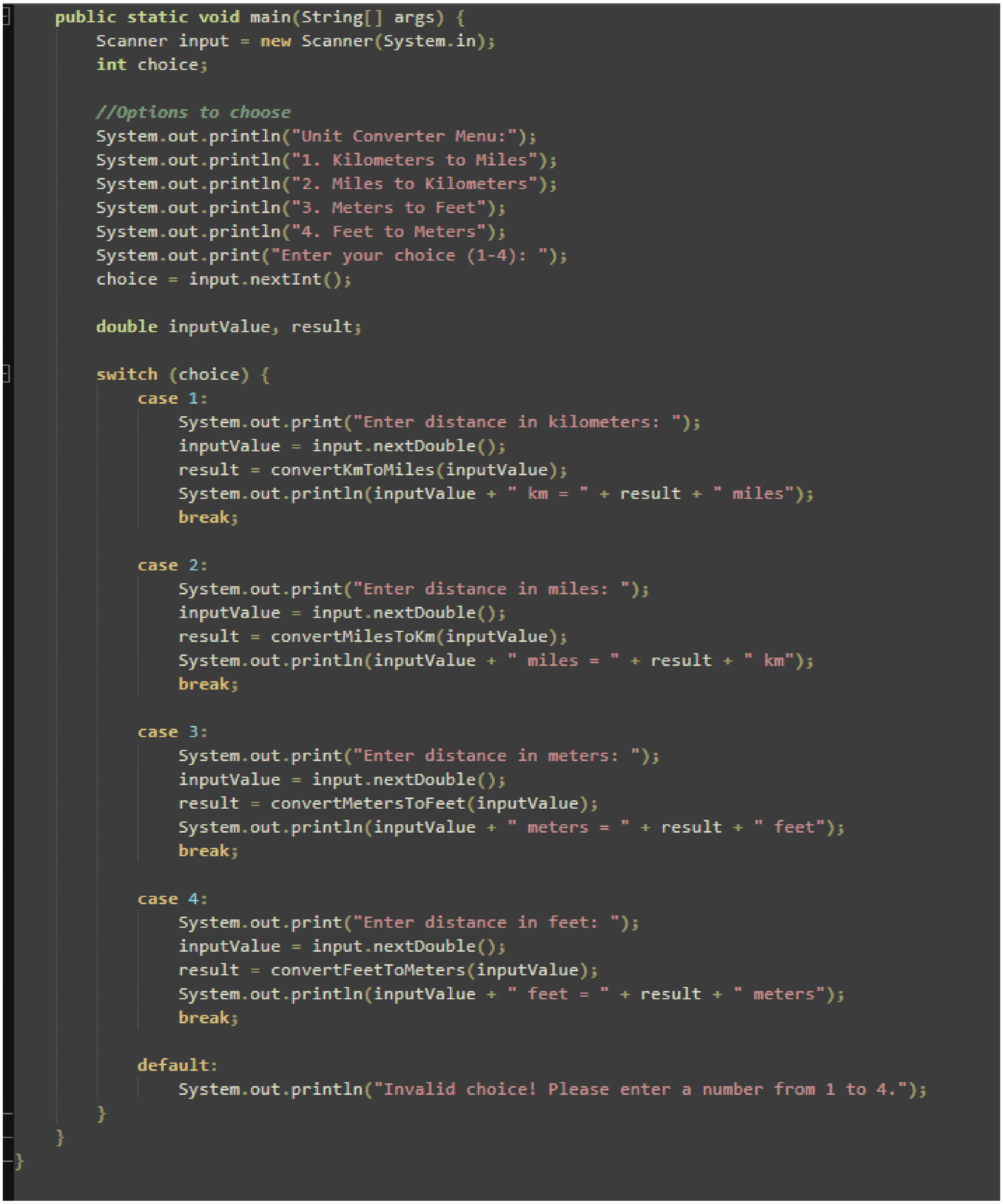


```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>javac LeapYear.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>java LeapYear
Enter a year : 2025
2025 is not a Leap Year.
```

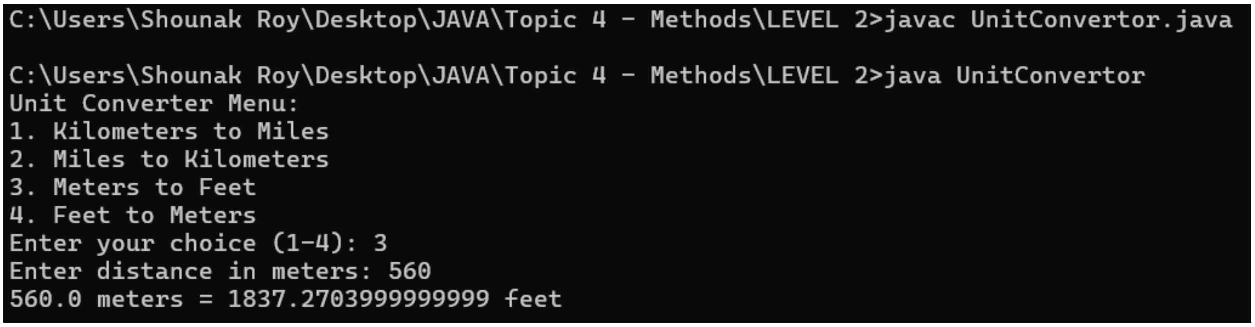
4. **Extend or Create a *UnitConvertor* utility class similar to the one shown in the notes to do the following. Please define *static* methods for all the UnitConvertor class methods.**



```java
import java.util.Scanner;
public class UnitConvertor {

    //Convert kilometers to miles
    public static double convertKmToMiles(double km) {
        double kmTOmiles = 0.621371;
        return km * kmTOmiles;
    }

    //Convert miles to kilometers
    public static double convertMilesToKm(double miles) {
        double milesTOkm = 1.60934;
        return miles * milesTOkm;
    }

    //Convert meters to feet
    public static double convertMetersToFeet(double meters) {
        double metersTOfeet = 3.28084;
        return meters * metersTOfeet;
    }

    //Convert feet to meters
    public static double convertFeetToMeters(double feet) {
        double feetTOmeters = 0.3048;
        return feet * feetTOmeters;
    }
```

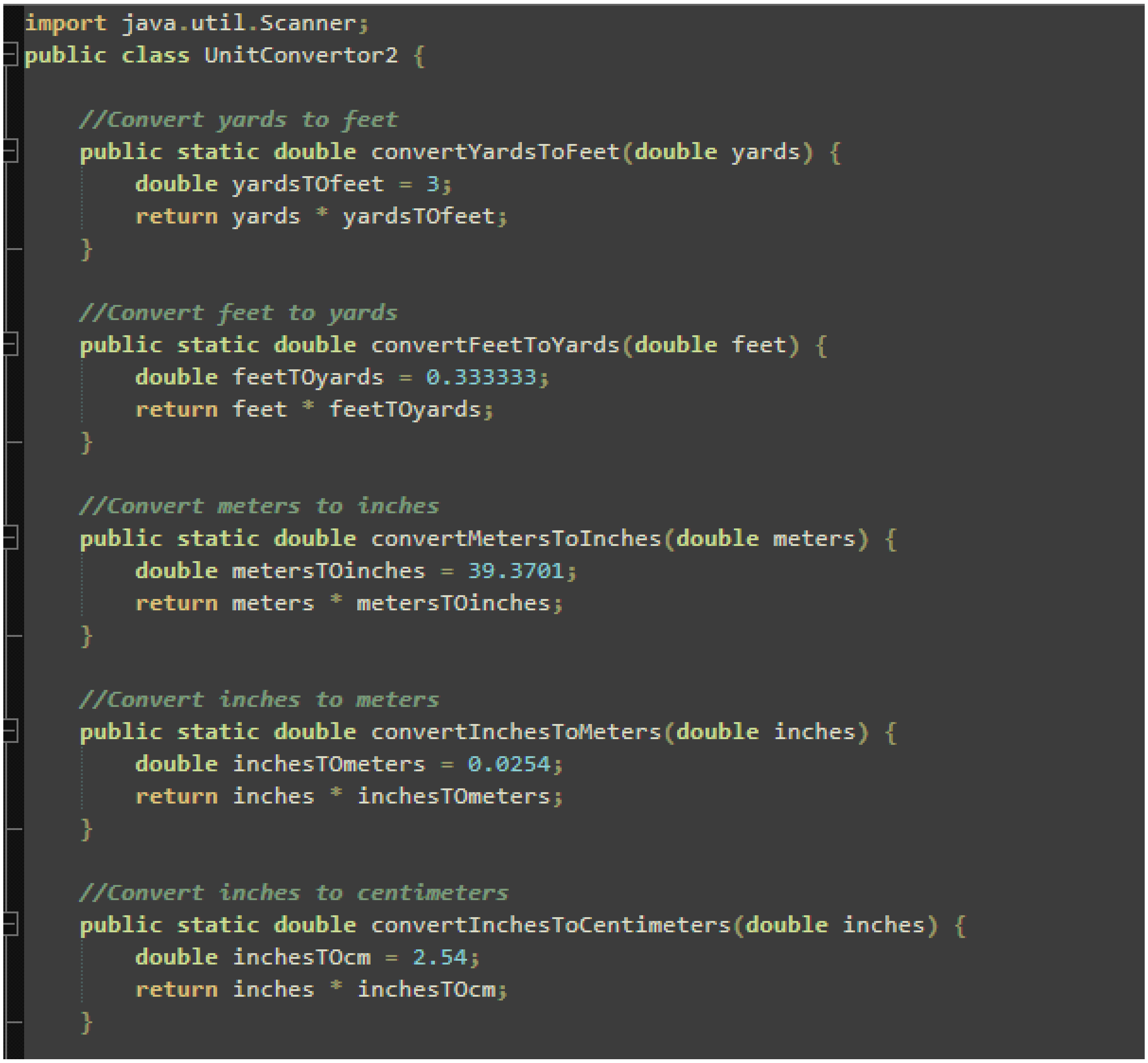


```java
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int choice;

        //Options to choose
        System.out.println("Unit Converter Menu:");
        System.out.println("1. Kilometers to Miles");
        System.out.println("2. Miles to Kilometers");
        System.out.println("3. Meters to Feet");
        System.out.println("4. Feet to Meters");
        System.out.print("Enter your choice (1-4): ");
        choice = input.nextInt();

        double inputValue, result;

        switch (choice) {
            case 1:
                System.out.print("Enter distance in kilometers: ");
                inputValue = input.nextDouble();
                result = convertKmToMiles(inputValue);
                System.out.println(inputValue + " km = " + result + " miles");
                break;

            case 2:
                System.out.print("Enter distance in miles: ");
                inputValue = input.nextDouble();
                result = convertMilesToKm(inputValue);
                System.out.println(inputValue + " miles = " + result + " km");
                break;

            case 3:
                System.out.print("Enter distance in meters: ");
                inputValue = input.nextDouble();
                result = convertMetersToFeet(inputValue);
                System.out.println(inputValue + " meters = " + result + " feet");
                break;

            case 4:
                System.out.print("Enter distance in feet: ");
                inputValue = input.nextDouble();
                result = convertFeetToMeters(inputValue);
                System.out.println(inputValue + " feet = " + result + " meters");
                break;

            default:
                System.out.println("Invalid choice! Please enter a number from 1 to 4.");
        }
    }
}
```

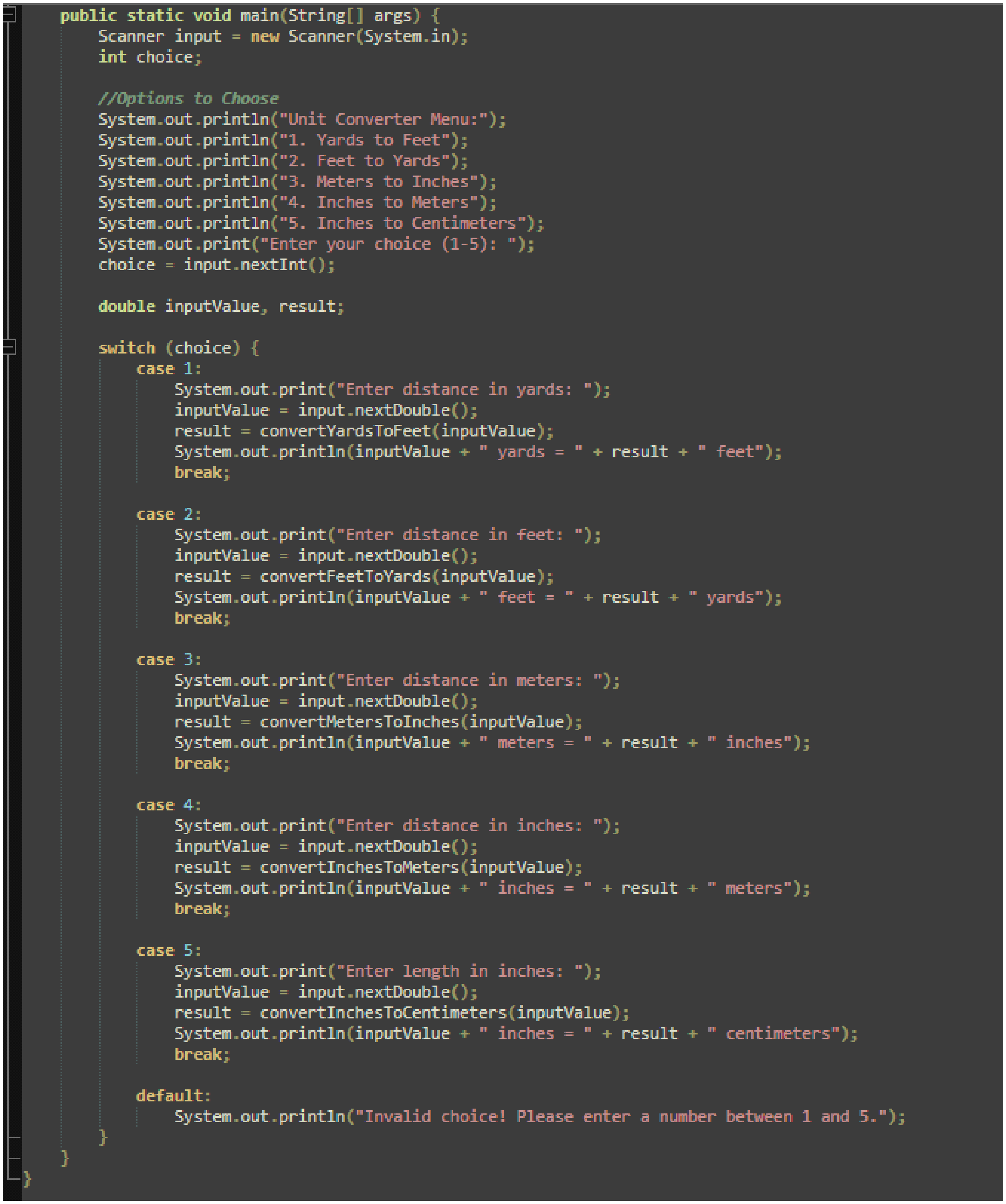


```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>javac UnitConvertor.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>java UnitConvertor
Unit Converter Menu:
1. Kilometers to Miles
2. Miles to Kilometers
3. Meters to Feet
4. Feet to Meters
Enter your choice (1-4): 3
Enter distance in meters: 560
560.0 meters = 1837.2703999999999 feet
```

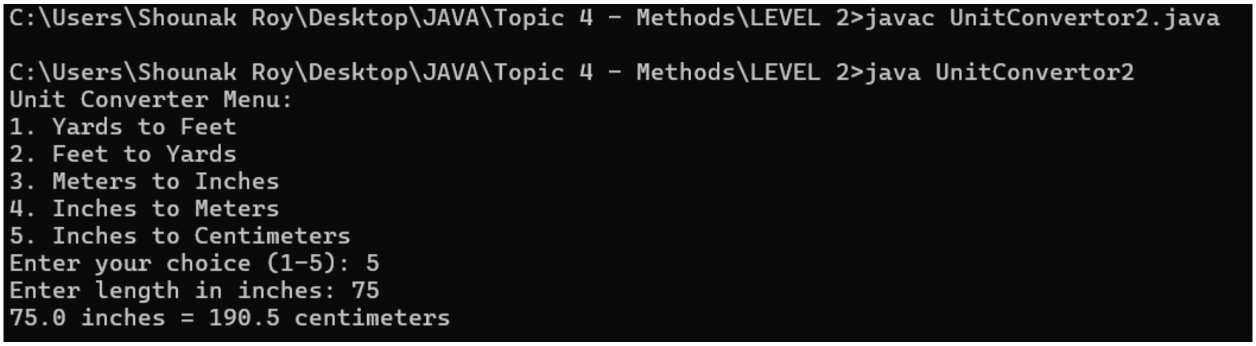
5. **Extend or Create a *UnitConvertor* utility class similar to the one shown in the notes to do the following. Please define *static* methods for all the UnitConvertor class methods.**



```
import java.util.Scanner;
public class UnitConvertor2 {

    //Convert yards to feet
    public static double convertYardsToFeet(double yards) {
        double yardsTOfeet = 3;
        return yards * yardsTOfeet;
    }

    //Convert feet to yards
    public static double convertFeetToYards(double feet) {
        double feetTOyards = 0.333333;
        return feet * feetTOyards;
    }

    //Convert meters to inches
    public static double convertMetersToInches(double meters) {
        double metersTOinches = 39.3701;
        return meters * metersTOinches;
    }

    //Convert inches to meters
    public static double convertInchesToMeters(double inches) {
        double inchesTOmeters = 0.0254;
        return inches * inchesTOmeters;
    }

    //Convert inches to centimeters
    public static double convertInchesToCentimeters(double inches) {
        double inchesTOcm = 2.54;
        return inches * inchesTOcm;
    }
```

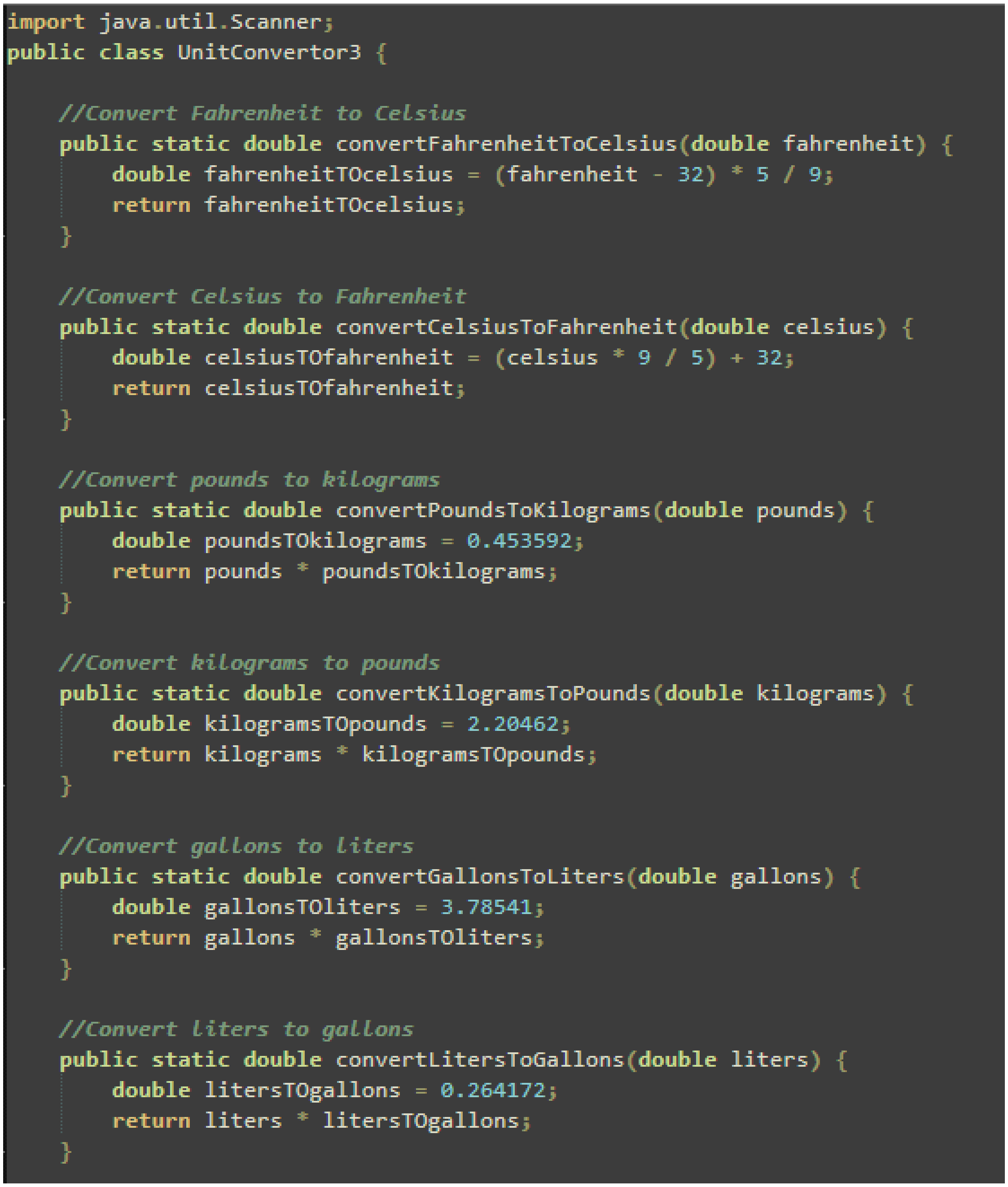


```java
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int choice;

        //Options to Choose
        System.out.println("Unit Converter Menu:");
        System.out.println("1. Yards to Feet");
        System.out.println("2. Feet to Yards");
        System.out.println("3. Meters to Inches");
        System.out.println("4. Inches to Meters");
        System.out.println("5. Inches to Centimeters");
        System.out.print("Enter your choice (1-5): ");
        choice = input.nextInt();

        double inputValue, result;

        switch (choice) {
            case 1:
                System.out.print("Enter distance in yards: ");
                inputValue = input.nextDouble();
                result = convertYardsToFeet(inputValue);
                System.out.println(inputValue + " yards = " + result + " feet");
                break;

            case 2:
                System.out.print("Enter distance in feet: ");
                inputValue = input.nextDouble();
                result = convertFeetToYards(inputValue);
                System.out.println(inputValue + " feet = " + result + " yards");
                break;

            case 3:
                System.out.print("Enter distance in meters: ");
                inputValue = input.nextDouble();
                result = convertMetersToInches(inputValue);
                System.out.println(inputValue + " meters = " + result + " inches");
                break;

            case 4:
                System.out.print("Enter distance in inches: ");
                inputValue = input.nextDouble();
                result = convertInchesToMeters(inputValue);
                System.out.println(inputValue + " inches = " + result + " meters");
                break;

            case 5:
                System.out.print("Enter length in inches: ");
                inputValue = input.nextDouble();
                result = convertInchesToCentimeters(inputValue);
                System.out.println(inputValue + " inches = " + result + " centimeters");
                break;

            default:
                System.out.println("Invalid choice! Please enter a number between 1 and 5.");
        }
    }
}
```

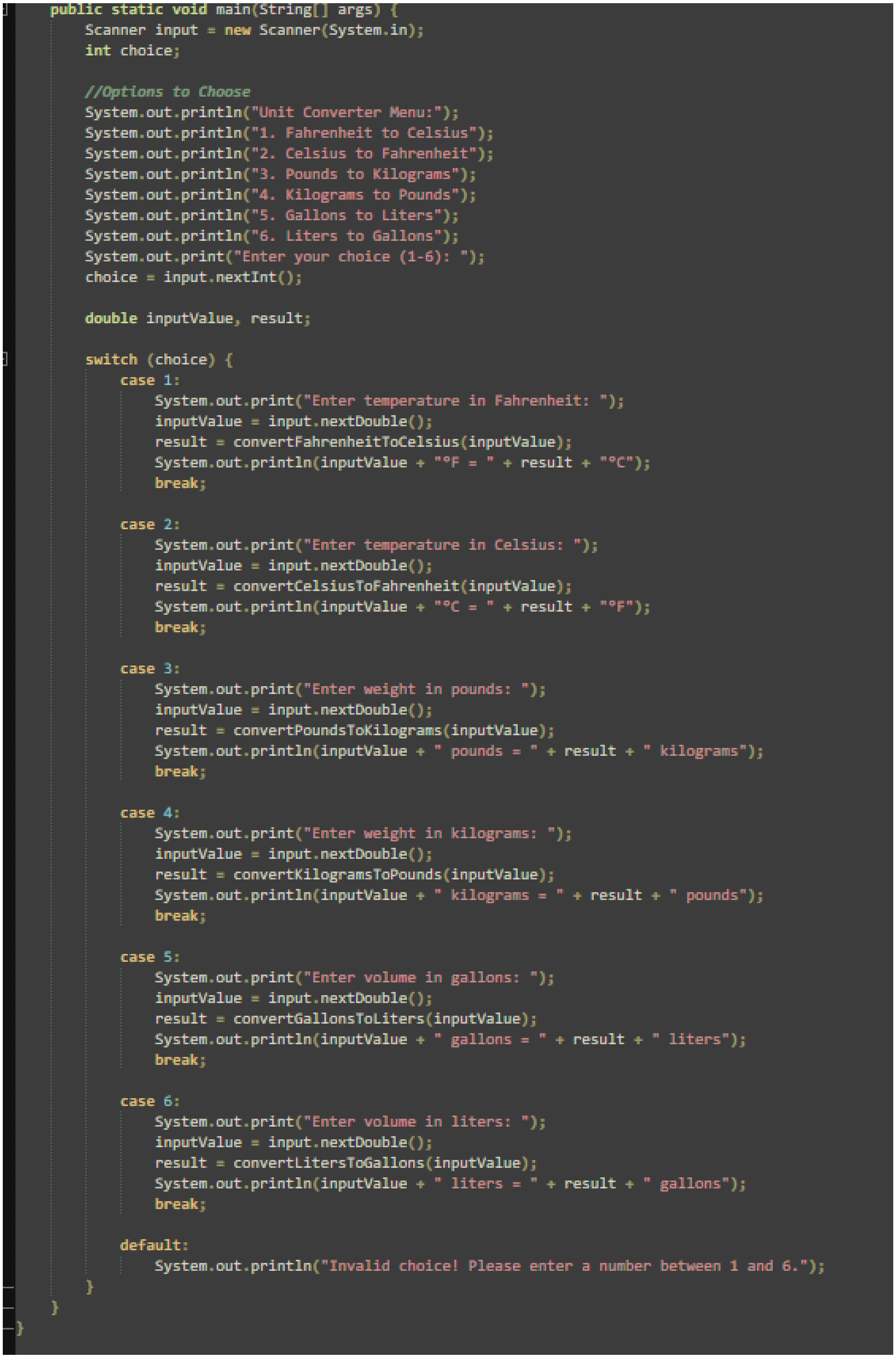


```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>javac UnitConvertor2.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>java UnitConvertor2
Unit Converter Menu:
1. Yards to Feet
2. Feet to Yards
3. Meters to Inches
4. Inches to Meters
5. Inches to Centimeters
Enter your choice (1-5): 5
Enter length in inches: 75
75.0 inches = 190.5 centimeters
```

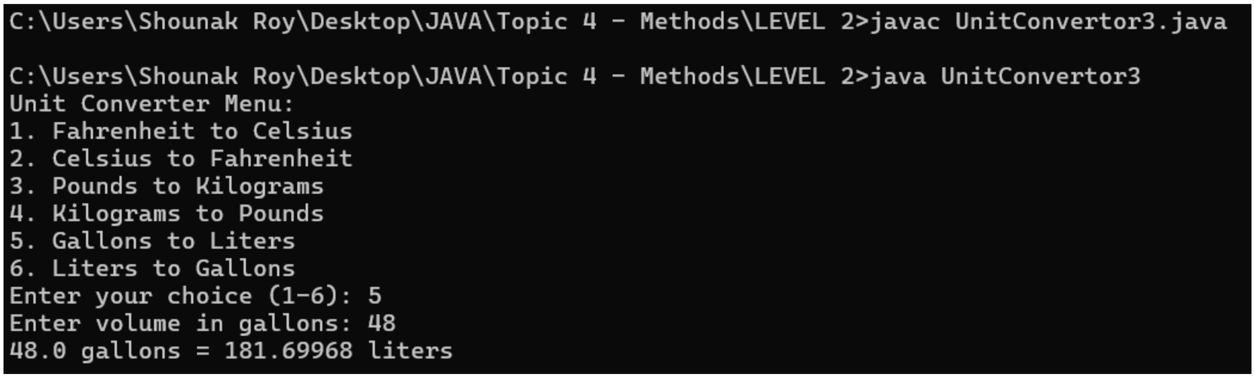
6. **Extend or Create a *UnitConvertor* utility class similar to the one shown in the notes to do the following. Please define *static* methods for all the UnitConvertor class methods.**



```java
import java.util.Scanner;
public class UnitConvertor3 {

    //Convert Fahrenheit to Celsius
    public static double convertFahrenheitToCelsius(double fahrenheit) {
        double fahrenheitTOcelsius = (fahrenheit - 32) * 5 / 9;
        return fahrenheitTOcelsius;
    }

    //Convert Celsius to Fahrenheit
    public static double convertCelsiusToFahrenheit(double celsius) {
        double celsiusTOfahrenheit = (celsius * 9 / 5) + 32;
        return celsiusTOfahrenheit;
    }

    //Convert pounds to kilograms
    public static double convertPoundsToKilograms(double pounds) {
        double poundsTOkilograms = 0.453592;
        return pounds * poundsTOkilograms;
    }

    //Convert kilograms to pounds
    public static double convertKilogramsToPounds(double kilograms) {
        double kilogramsTOpounds = 2.20462;
        return kilograms * kilogramsTOpounds;
    }

    //Convert gallons to liters
    public static double convertGallonsToLiters(double gallons) {
        double gallonsTOliters = 3.78541;
        return gallons * gallonsTOliters;
    }

    //Convert liters to gallons
    public static double convertLitersToGallons(double liters) {
        double litersTOgallons = 0.264172;
        return liters * litersTOgallons;
    }
```

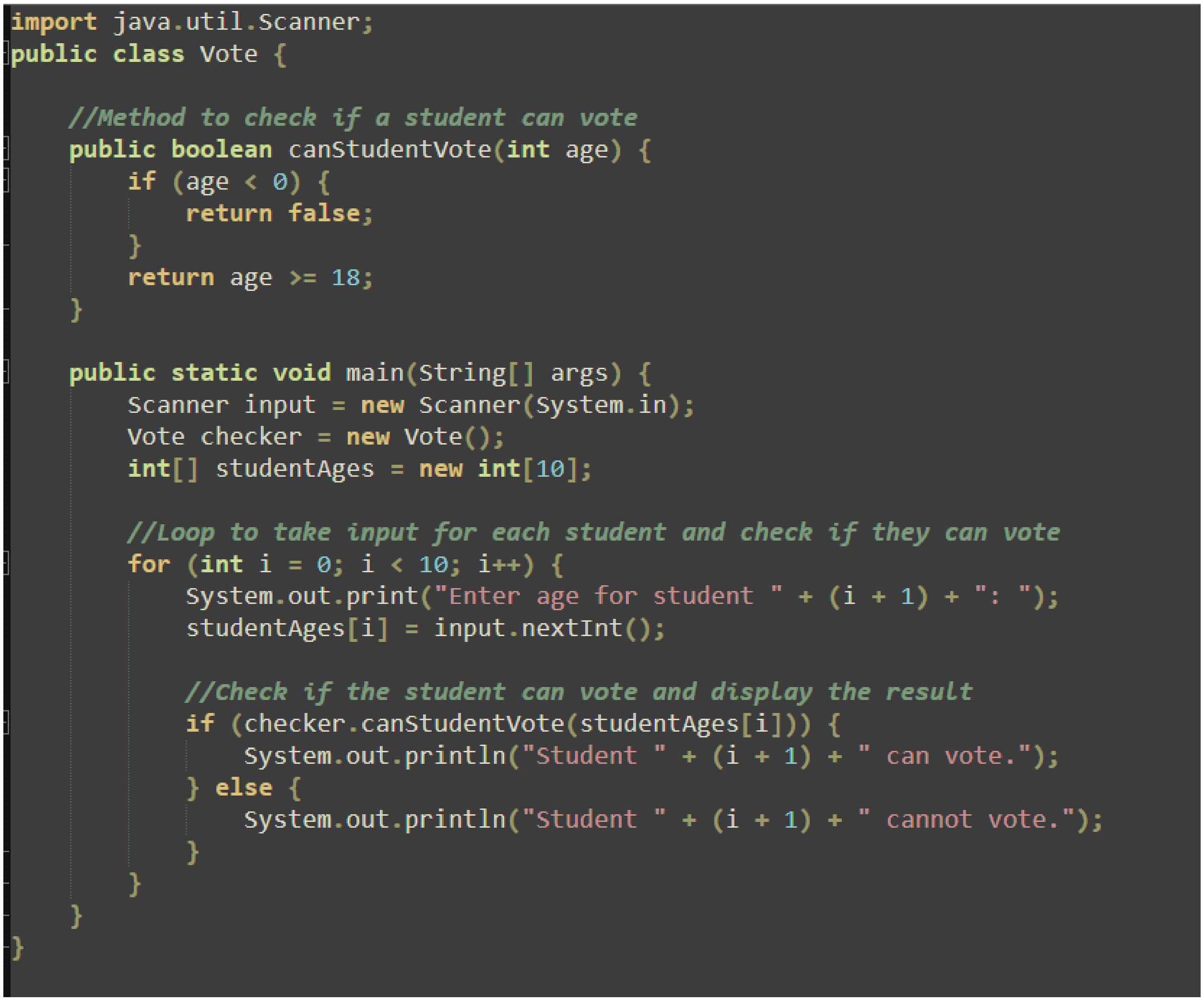


```
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int choice;

        //Options to Choose
        System.out.println("Unit Converter Menu:");
        System.out.println("1. Fahrenheit to Celsius");
        System.out.println("2. Celsius to Fahrenheit");
        System.out.println("3. Pounds to Kilograms");
        System.out.println("4. Kilograms to Pounds");
        System.out.println("5. Gallons to Liters");
        System.out.println("6. Liters to Gallons");
        System.out.print("Enter your choice (1-6): ");
        choice = input.nextInt();

        double inputValue, result;

        switch (choice) {
            case 1:
                System.out.print("Enter temperature in Fahrenheit: ");
                inputValue = input.nextDouble();
                result = convertFahrenheitToCelsius(inputValue);
                System.out.println(inputValue + "°F = " + result + "°C");
                break;

            case 2:
                System.out.print("Enter temperature in Celsius: ");
                inputValue = input.nextDouble();
                result = convertCelsiusToFahrenheit(inputValue);
                System.out.println(inputValue + "°C = " + result + "°F");
                break;

            case 3:
                System.out.print("Enter weight in pounds: ");
                inputValue = input.nextDouble();
                result = convertPoundsToKilograms(inputValue);
                System.out.println(inputValue + " pounds = " + result + " kilograms");
                break;

            case 4:
                System.out.print("Enter weight in kilograms: ");
                inputValue = input.nextDouble();
                result = convertKilogramsToPounds(inputValue);
                System.out.println(inputValue + " kilograms = " + result + " pounds");
                break;

            case 5:
                System.out.print("Enter volume in gallons: ");
                inputValue = input.nextDouble();
                result = convertGallonsToLiters(inputValue);
                System.out.println(inputValue + " gallons = " + result + " liters");
                break;

            case 6:
                System.out.print("Enter volume in liters: ");
                inputValue = input.nextDouble();
                result = convertLitersToGallons(inputValue);
                System.out.println(inputValue + " liters = " + result + " gallons");
                break;

            default:
                System.out.println("Invalid choice! Please enter a number between 1 and 6.");
        }
    }
}
```

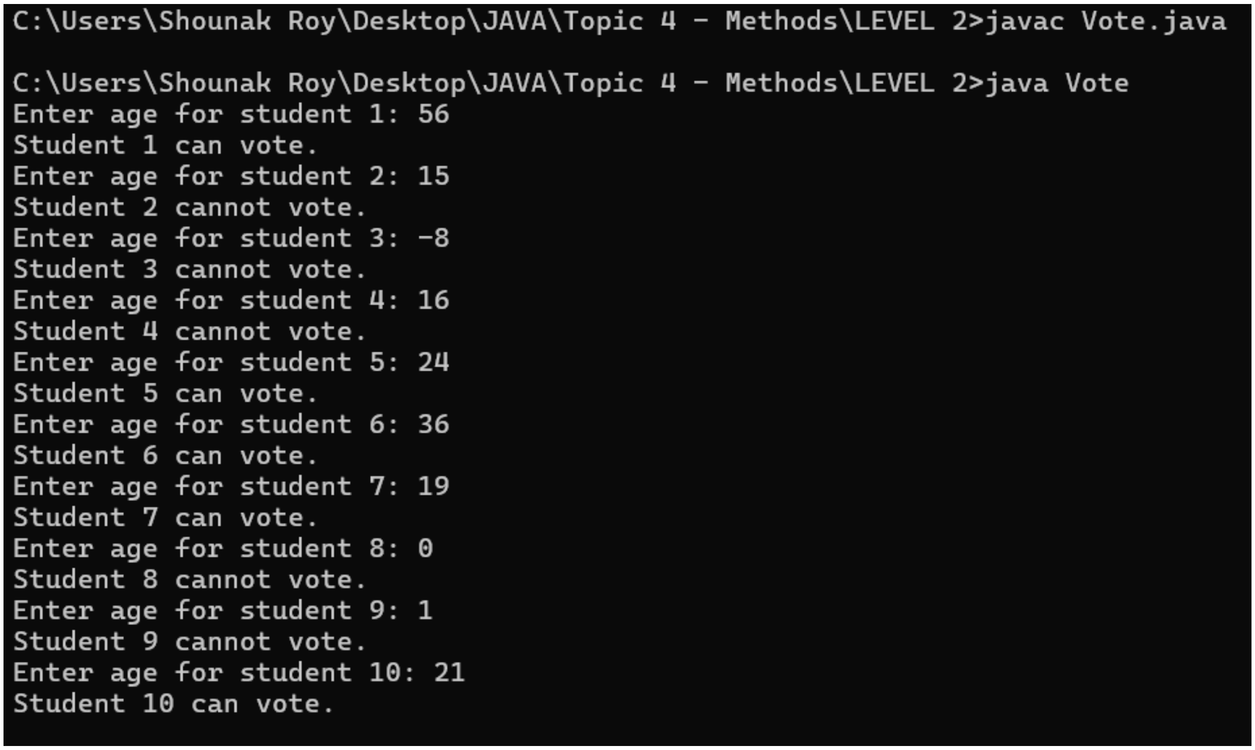


```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>javac UnitConvertor3.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>java UnitConvertor3
Unit Converter Menu:
1. Fahrenheit to Celsius
2. Celsius to Fahrenheit
3. Pounds to Kilograms
4. Kilograms to Pounds
5. Gallons to Liters
6. Liters to Gallons
Enter your choice (1-6): 5
Enter volume in gallons: 48
48.0 gallons = 181.69968 liters
```

7. **Write a program to take user input for the age of all 10 students in a class and check whether the student can vote depending on his/her age is greater or equal to 18.**

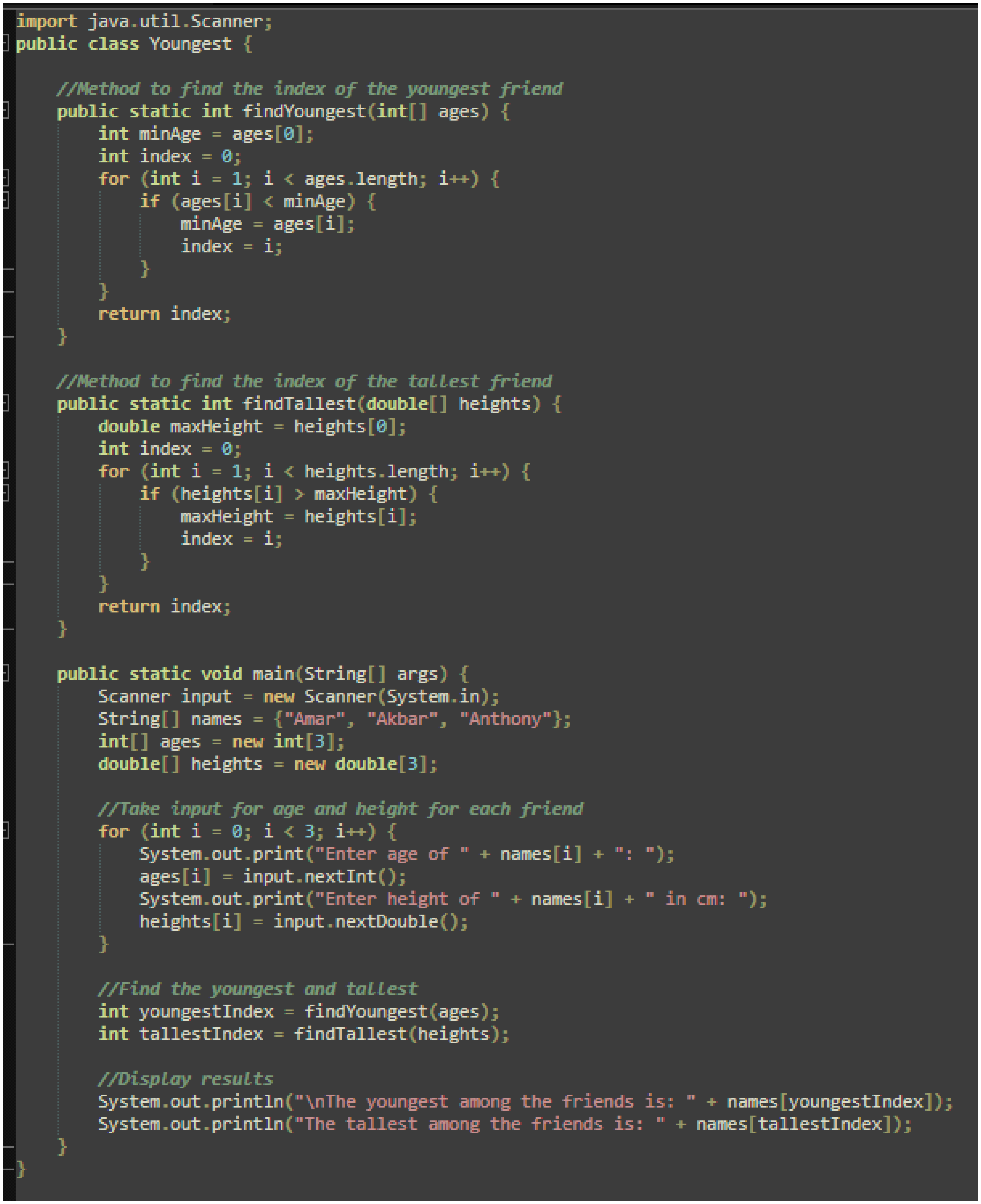


```java
import java.util.Scanner;
public class Vote {

    //Method to check if a student can vote
    public boolean canStudentVote(int age) {
        if (age < 0) {
            return false;
        }
        return age >= 18;
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        Vote checker = new Vote();
        int[] studentAges = new int[10];

        //Loop to take input for each student and check if they can vote
        for (int i = 0; i < 10; i++) {
            System.out.print("Enter age for student " + (i + 1) + ": ");
            studentAges[i] = input.nextInt();

            //Check if the student can vote and display the result
            if (checker.canStudentVote(studentAges[i])) {
                System.out.println("Student " + (i + 1) + " can vote.");
            } else {
                System.out.println("Student " + (i + 1) + " cannot vote.");
            }
        }
    }
}
```

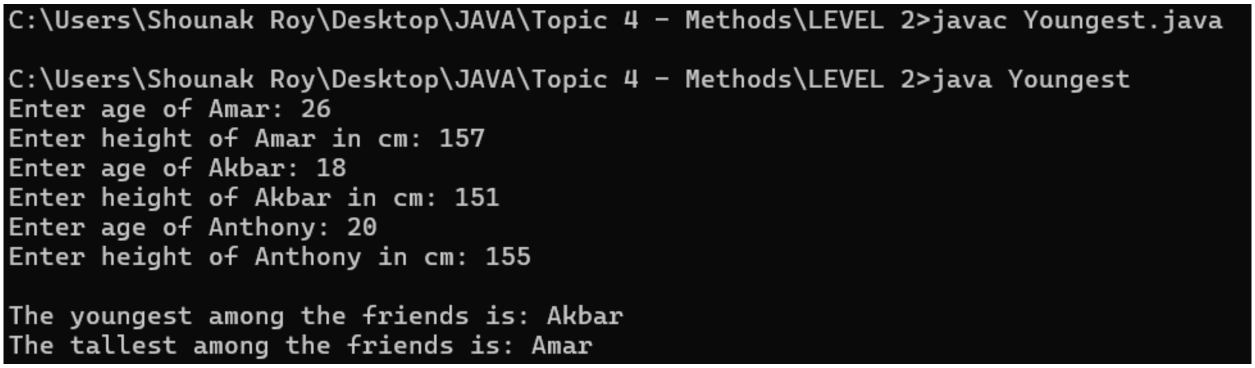


```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>javac Vote.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>java Vote
Enter age for student 1: 56
Student 1 can vote.
Enter age for student 2: 15
Student 2 cannot vote.
Enter age for student 3: -8
Student 3 cannot vote.
Enter age for student 4: 16
Student 4 cannot vote.
Enter age for student 5: 24
Student 5 can vote.
Enter age for student 6: 36
Student 6 can vote.
Enter age for student 7: 19
Student 7 can vote.
Enter age for student 8: 0
Student 8 cannot vote.
Enter age for student 9: 1
Student 9 cannot vote.
Enter age for student 10: 21
Student 10 can vote.
```

8. **Create a program to find the youngest friends among 3 Amar, Akbar and Anthony based on their ages and tallest among the friends based on their heights and display it.**

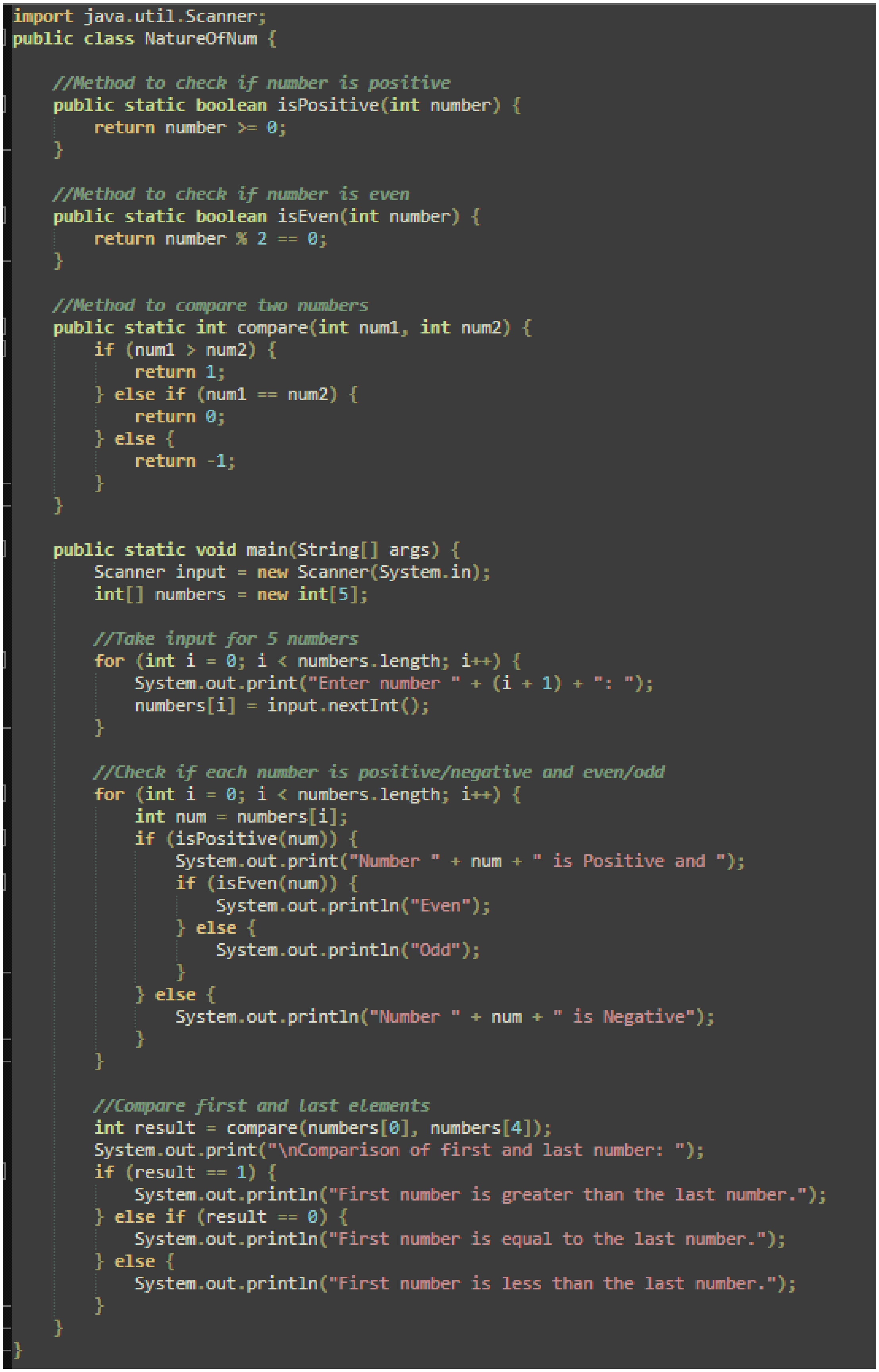


```java
import java.util.Scanner;
public class Youngest {

    //Method to find the index of the youngest friend
    public static int findYoungest(int[] ages) {
        int minAge = ages[0];
        int index = 0;
        for (int i = 1; i < ages.length; i++) {
            if (ages[i] < minAge) {
                minAge = ages[i];
                index = i;
            }
        }
        return index;
    }

    //Method to find the index of the tallest friend
    public static int findTallest(double[] heights) {
        double maxHeight = heights[0];
        int index = 0;
        for (int i = 1; i < heights.length; i++) {
            if (heights[i] > maxHeight) {
                maxHeight = heights[i];
                index = i;
            }
        }
        return index;
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        String[] names = {"Amar", "Akbar", "Anthony"};
        int[] ages = new int[3];
        double[] heights = new double[3];

        //Take input for age and height for each friend
        for (int i = 0; i < 3; i++) {
            System.out.print("Enter age of " + names[i] + ": ");
            ages[i] = input.nextInt();
            System.out.print("Enter height of " + names[i] + " in cm: ");
            heights[i] = input.nextDouble();
        }

        //Find the youngest and tallest
        int youngestIndex = findYoungest(ages);
        int tallestIndex = findTallest(heights);

        //Display results
        System.out.println("\nThe youngest among the friends is: " + names[youngestIndex]);
        System.out.println("The tallest among the friends is: " + names[tallestIndex]);
    }
}
```

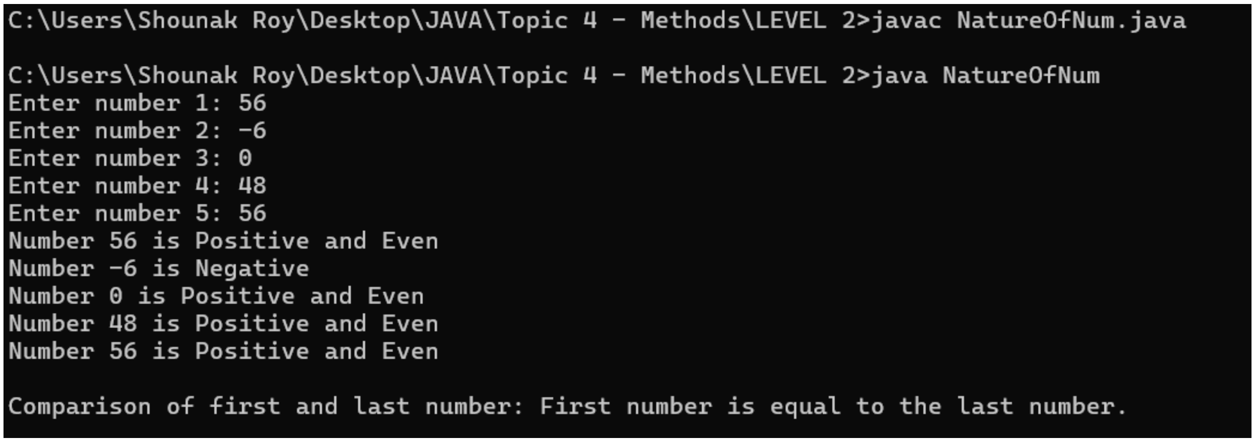


```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>javac Youngest.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>java Youngest
Enter age of Amar: 26
Enter height of Amar in cm: 157
Enter age of Akbar: 18
Enter height of Akbar in cm: 151
Enter age of Anthony: 20
Enter height of Anthony in cm: 155

The youngest among the friends is: Akbar
The tallest among the friends is: Amar
```

9. **Write a program to take user input for 5 numbers and check whether a number is positive or negative. Further for positive numbers check if the number is even or odd. Finally compare the first and last elements of the array and display if they are equal, greater, or less.**

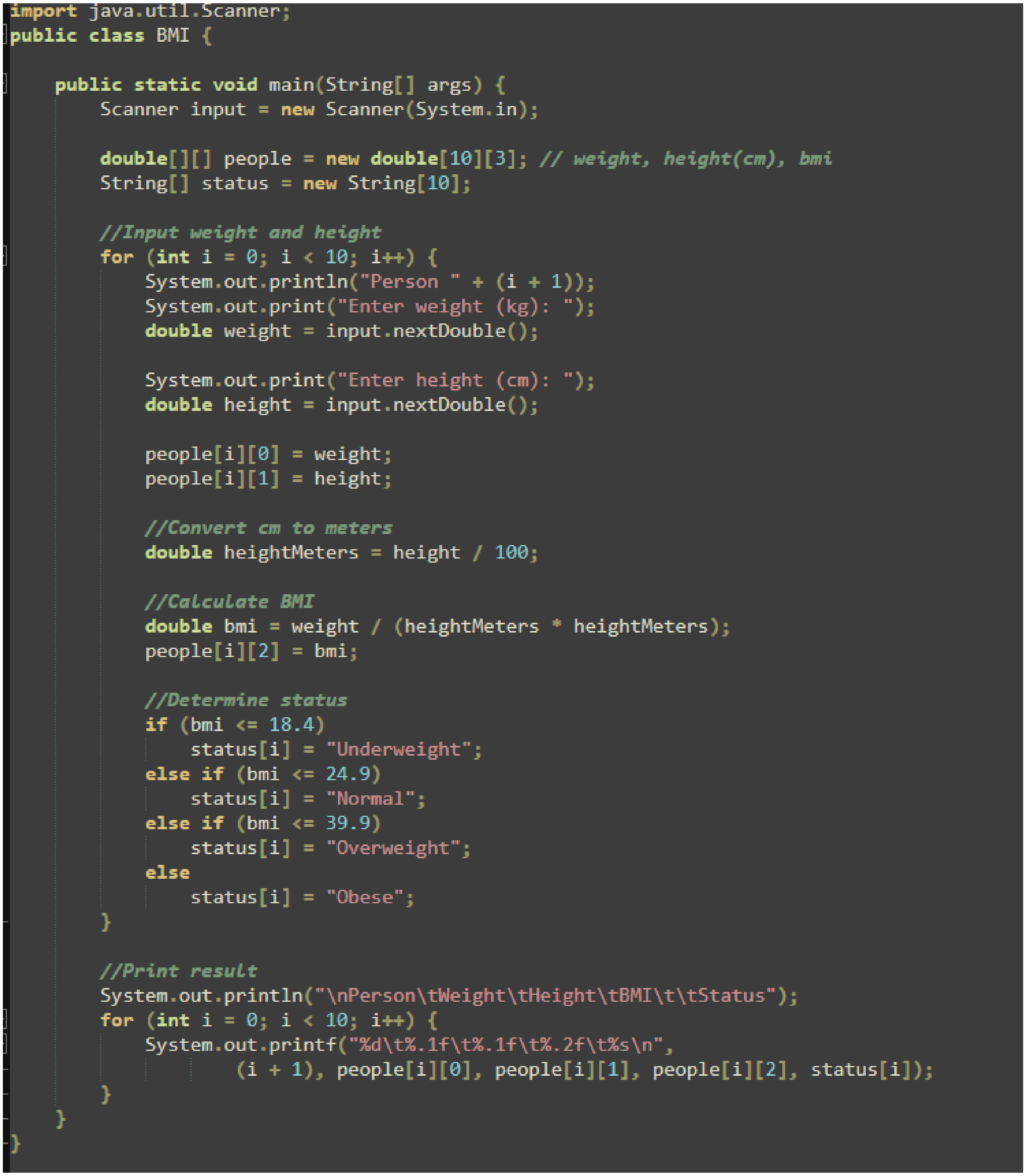


```java
import java.util.Scanner;
public class NatureOfNum {

    //Method to check if number is positive
    public static boolean isPositive(int number) {
        return number >= 0;
    }

    //Method to check if number is even
    public static boolean isEven(int number) {
        return number % 2 == 0;
    }

    //Method to compare two numbers
    public static int compare(int num1, int num2) {
        if (num1 > num2) {
            return 1;
        } else if (num1 == num2) {
            return 0;
        } else {
            return -1;
        }
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int[] numbers = new int[5];

        //Take input for 5 numbers
        for (int i = 0; i < numbers.length; i++) {
            System.out.print("Enter number " + (i + 1) + ": ");
            numbers[i] = input.nextInt();
        }

        //Check if each number is positive/negative and even/odd
        for (int i = 0; i < numbers.length; i++) {
            int num = numbers[i];
            if (isPositive(num)) {
                System.out.print("Number " + num + " is Positive and ");
                if (isEven(num)) {
                    System.out.println("Even");
                } else {
                    System.out.println("Odd");
                }
            } else {
                System.out.println("Number " + num + " is Negative");
            }
        }

        //Compare first and last elements
        int result = compare(numbers[0], numbers[4]);
        System.out.print("\nComparison of first and last number: ");
        if (result == 1) {
            System.out.println("First number is greater than the last number.");
        } else if (result == 0) {
            System.out.println("First number is equal to the last number.");
        } else {
            System.out.println("First number is less than the last number.");
        }
    }
}
```

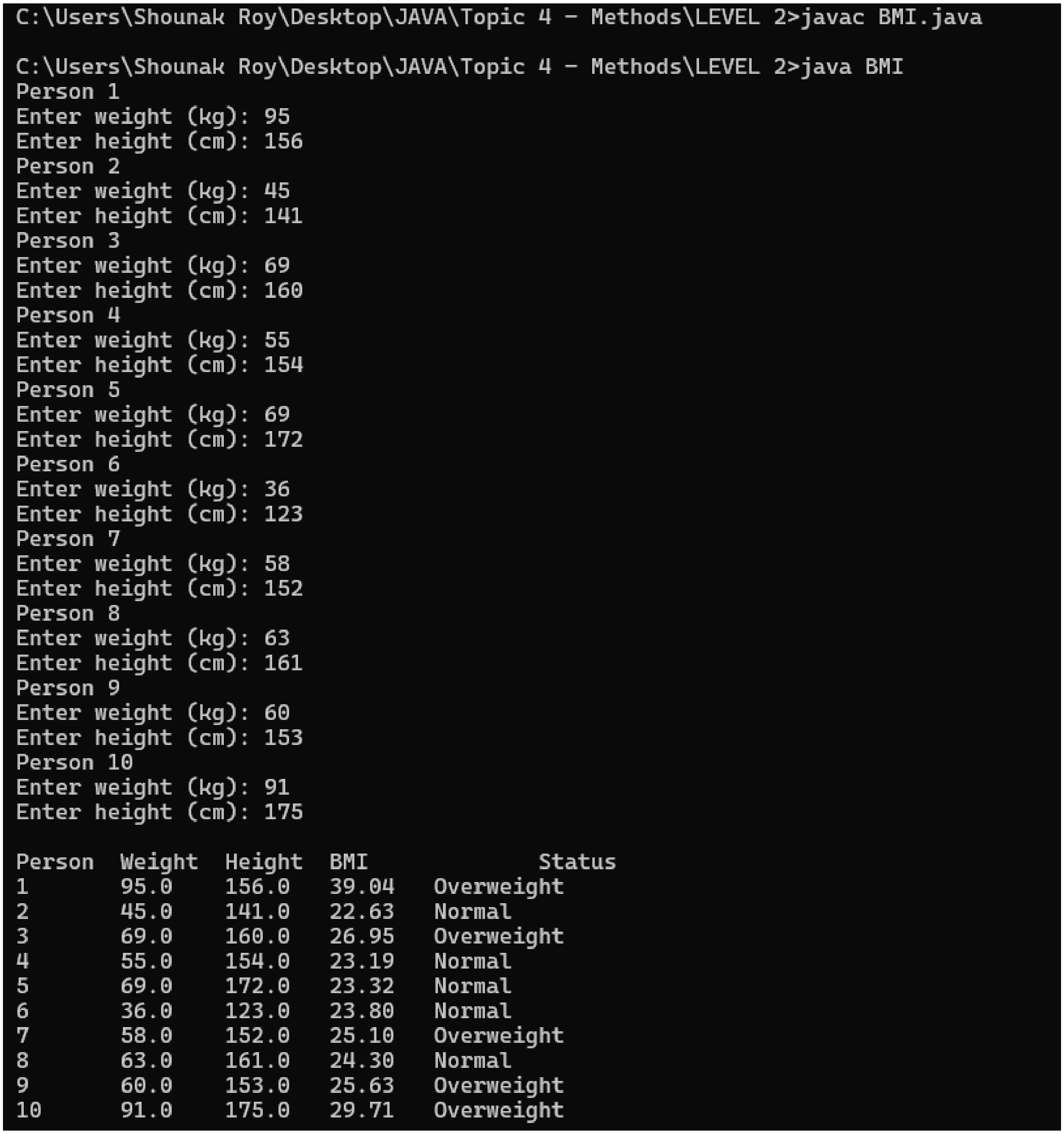


```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>javac NatureOfNum.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>java NatureOfNum
Enter number 1: 56
Enter number 2: -6
Enter number 3: 0
Enter number 4: 48
Enter number 5: 56
Number 56 is Positive and Even
Number -6 is Negative
Number 0 is Positive and Even
Number 48 is Positive and Even
Number 56 is Positive and Even

Comparison of first and last number: First number is equal to the last number.
```

10. An organization took up the exercise to find the Body Mass Index (BMI) of all the persons in the team of 10 members. For this create a program to find the BMI and display the height, weight, BMI and status of each individual.

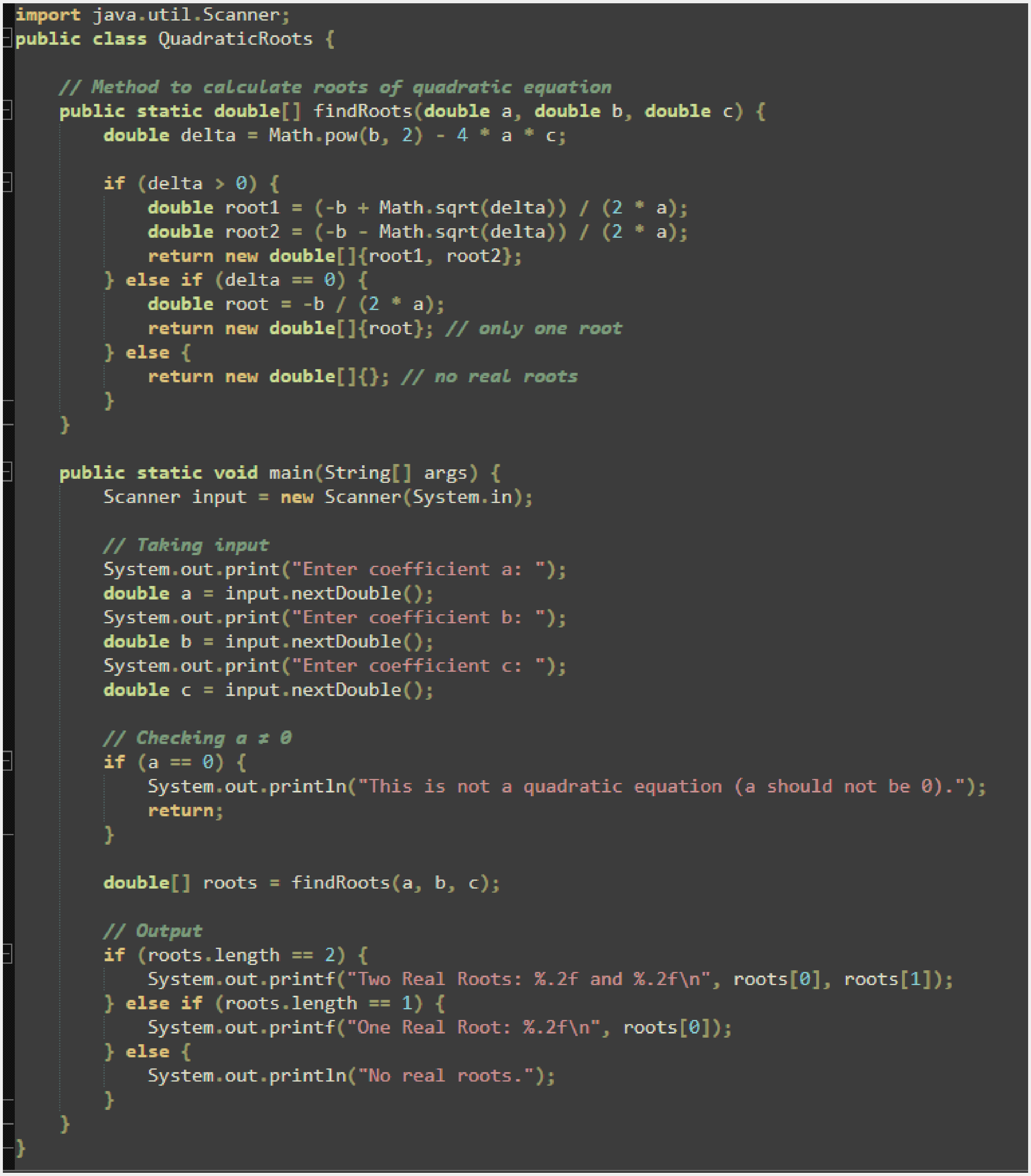


```java
import java.util.Scanner;
public class BMI {

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        double[][] people = new double[10][3]; // weight, height(cm), bmi
        String[] status = new String[10];

        //Input weight and height
        for (int i = 0; i < 10; i++) {
            System.out.println("Person " + (i + 1));
            System.out.print("Enter weight (kg): ");
            double weight = input.nextDouble();

            System.out.print("Enter height (cm): ");
            double height = input.nextDouble();

            people[i][0] = weight;
            people[i][1] = height;

            //Convert cm to meters
            double heightMeters = height / 100;

            //Calculate BMI
            double bmi = weight / (heightMeters * heightMeters);
            people[i][2] = bmi;

            //Determine status
            if (bmi <= 18.4)
                status[i] = "Underweight";
            else if (bmi <= 24.9)
                status[i] = "Normal";
            else if (bmi <= 39.9)
                status[i] = "Overweight";
            else
                status[i] = "Obese";
        }

        //Print result
        System.out.println("\nPerson\tWeight\tHeight\tBMI\t\tStatus");
        for (int i = 0; i < 10; i++) {
            System.out.printf("%d\t%.1f\t%.1f\t%.2f\t%s\n",
                    (i + 1), people[i][0], people[i][1], people[i][2], status[i]);
        }
    }
}
```

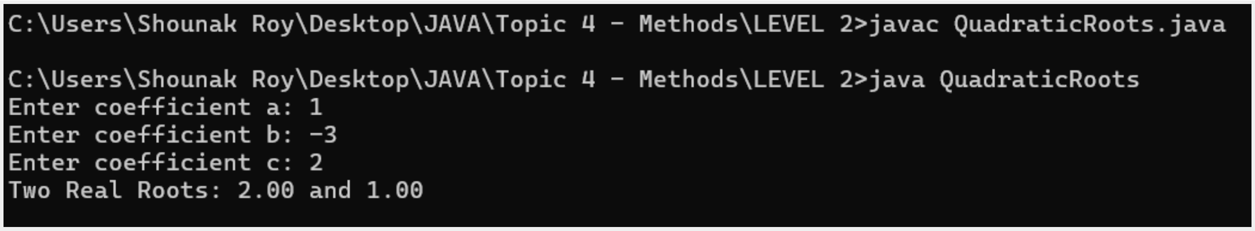


```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>javac BMI.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>java BMI
Person 1
Enter weight (kg): 95
Enter height (cm): 156
Person 2
Enter weight (kg): 45
Enter height (cm): 141
Person 3
Enter weight (kg): 69
Enter height (cm): 160
Person 4
Enter weight (kg): 55
Enter height (cm): 154
Person 5
Enter weight (kg): 69
Enter height (cm): 172
Person 6
Enter weight (kg): 36
Enter height (cm): 123
Person 7
Enter weight (kg): 58
Enter height (cm): 152
Person 8
Enter weight (kg): 63
Enter height (cm): 161
Person 9
Enter weight (kg): 60
Enter height (cm): 153
Person 10
Enter weight (kg): 91
Enter height (cm): 175

Person  Weight  Height  BMI             Status
1       95.0    156.0   39.04   Overweight
2       45.0    141.0   22.63   Normal
3       69.0    160.0   26.95   Overweight
4       55.0    154.0   23.19   Normal
5       69.0    172.0   23.32   Normal
6       36.0    123.0   23.80   Normal
7       58.0    152.0   25.10   Overweight
8       63.0    161.0   24.30   Normal
9       60.0    153.0   25.63   Overweight
10      91.0    175.0   29.71   Overweight
```

11. **Write a program Quadratic to find the roots of the equation $ax^2 + bx + c$. Use Math functions *Math.pow()* and *Math.sqrt().***

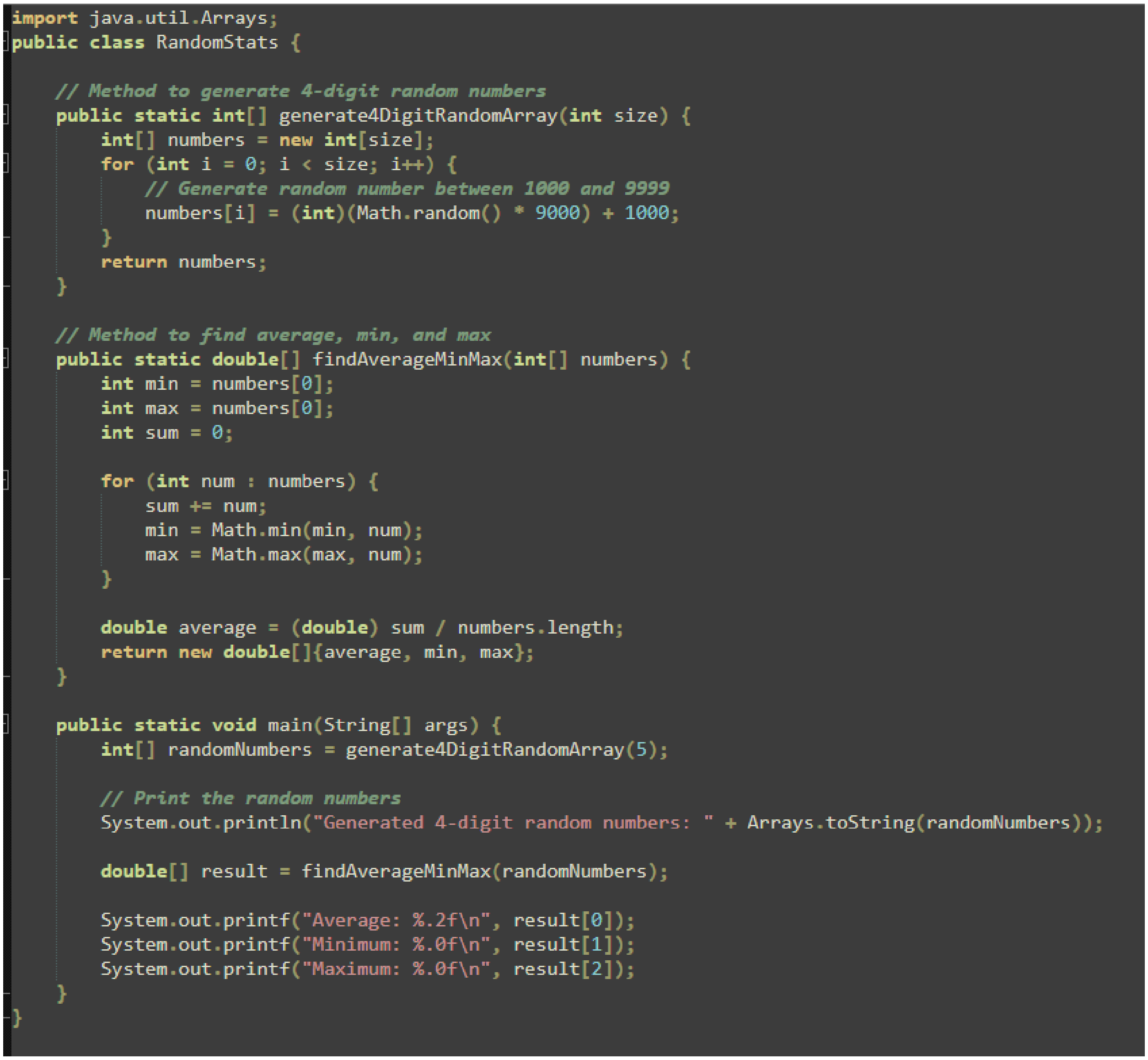


```java
import java.util.Scanner;
public class QuadraticRoots {

    // Method to calculate roots of quadratic equation
    public static double[] findRoots(double a, double b, double c) {
        double delta = Math.pow(b, 2) - 4 * a * c;

        if (delta > 0) {
            double root1 = (-b + Math.sqrt(delta)) / (2 * a);
            double root2 = (-b - Math.sqrt(delta)) / (2 * a);
            return new double[]{root1, root2};
        } else if (delta == 0) {
            double root = -b / (2 * a);
            return new double[]{root}; // only one root
        } else {
            return new double[]{}; // no real roots
        }
    }

    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        // Taking input
        System.out.print("Enter coefficient a: ");
        double a = input.nextDouble();
        System.out.print("Enter coefficient b: ");
        double b = input.nextDouble();
        System.out.print("Enter coefficient c: ");
        double c = input.nextDouble();

        // Checking a ≠ 0
        if (a == 0) {
            System.out.println("This is not a quadratic equation (a should not be 0).");
            return;
        }

        double[] roots = findRoots(a, b, c);

        // Output
        if (roots.length == 2) {
            System.out.printf("Two Real Roots: %.2f and %.2f\n", roots[0], roots[1]);
        } else if (roots.length == 1) {
            System.out.printf("One Real Root: %.2f\n", roots[0]);
        } else {
            System.out.println("No real roots.");
        }
    }
}
```

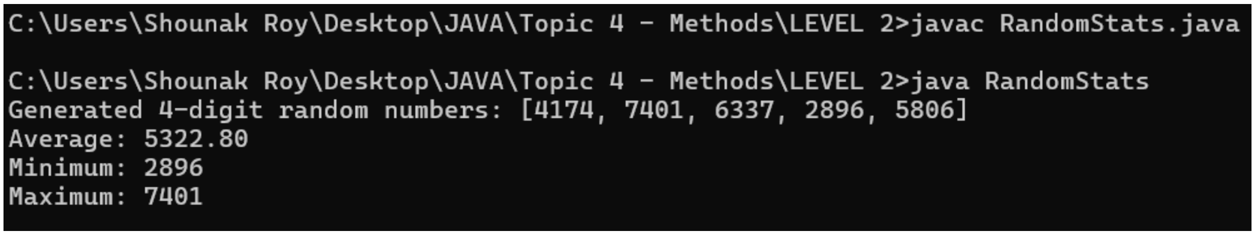


```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>javac QuadraticRoots.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>java QuadraticRoots
Enter coefficient a: 1
Enter coefficient b: -3
Enter coefficient c: 2
Two Real Roots: 2.00 and 1.00
```

12. **Write a program that generates five 4 digit random values and then finds their average value, and their minimum and maximum value. Use Math.random(), Math.min(), and Math.max().**



```java
import java.util.Arrays;
public class RandomStats {

    // Method to generate 4-digit random numbers
    public static int[] generate4DigitRandomArray(int size) {
        int[] numbers = new int[size];
        for (int i = 0; i < size; i++) {
            // Generate random number between 1000 and 9999
            numbers[i] = (int)(Math.random() * 9000) + 1000;
        }
        return numbers;
    }

    // Method to find average, min, and max
    public static double[] findAverageMinMax(int[] numbers) {
        int min = numbers[0];
        int max = numbers[0];
        int sum = 0;

        for (int num : numbers) {
            sum += num;
            min = Math.min(min, num);
            max = Math.max(max, num);
        }

        double average = (double) sum / numbers.length;
        return new double[]{average, min, max};
    }

    public static void main(String[] args) {
        int[] randomNumbers = generate4DigitRandomArray(5);

        // Print the random numbers
        System.out.println("Generated 4-digit random numbers: " + Arrays.toString(randomNumbers));

        double[] result = findAverageMinMax(randomNumbers);

        System.out.printf("Average: %.2f\n", result[0]);
        System.out.printf("Minimum: %.0f\n", result[1]);
        System.out.printf("Maximum: %.0f\n", result[2]);
    }
}
```



```
C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>javac RandomStats.java

C:\Users\Shounak Roy\Desktop\JAVA\Topic 4 - Methods\LEVEL 2>java RandomStats
Generated 4-digit random numbers: [4174, 7401, 6337, 2896, 5806]
Average: 5322.80
Minimum: 2896
Maximum: 7401
```