CodInClub

Powered By BridgeLabz

# Best Programming Practice

1. All values as variables including Fixed, User Inputs, and Results
2. Proper naming conventions for all variables
3. Proper Program Name and Class Name
4. Proper Method Name which indicates action taking inputs and providing result

**Sample Program 1:** Create a program to find the sum of all the digits of a number given by a user using an array and display the sum.

a. Use Math.random() and get a 4-digit random integer number
b. Write a method to count digits in the number
c. Write a method to return an array of digits from a given number.
d. Write a method to Find the sum of the digits of the number in the array
e. Finally, display the sum of the digits of the number

Java

```java
// Create SumOfDigit Class to compute the sum of 4 digits random number
class SumOfDigits {
   // Get a 4 digit random number
   public int get4DigitRandomNumber() {
      return (int) (Math.random() * 9000) + 1000;
   }

   // Find the count of digits in the number
   public int countDigits(int number) {
      int count = 0, temp = number;
      while (temp > 0) {
         count++;
         temp /= 10;
      }
      return count;
   }

   // Store the digits of the number in an array
   public int[] getDigits(int number, int count) {
      int[] digits = new int[count];
      int temp = number;
      for (int i = count - 1; i >= 0; i--) {
         digits[i] = temp % 10;
         temp /= 10;
      }
      return digits;
   }
```

```
    // Find the sum of the elements in an array
    public int sumArray(int[] array) {
        int sum = 0;
        for (int i = 0; i < array.length; i++) {
            sum += array[i];
        }
        return sum;
    }

    public static void main(String[] args) {
        // Get 4 digit random integer number
        SumOfDigits sumOfDigits = new SumOfDigits();
        int number = sumOfDigits.get4DigitRandomNumber();
        System.out.println("The Random Mumber is: " + number);

        // Get the count of digits
        int count = sumOfDigits.countDigits(number);
        System.out.println("The count of digit is: " + count);

        // Get the array of digits from the number
        int[] digits = sumOfDigits.getDigits(number, count);

        // Find the sum of the digits of the number
        int sum = sumOfDigits.sumArray(digits);

        // Display the sum of the digits of the number
        System.out.println("\nSum of Digits: " + sum);
    }
}
```

# 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

   **Hint =>**

   a. Take the input for a number
   b. Write a ***static*** Method to find the factors of the number and save them in an array and return the array.
   c. To find factors and save to array will have two loops. The first loop to find the count and initialize the array with the count. And the second loop save the factors into the array
   d. Write a method to find the sum of the factors using factors array
   e. Write a method to find the product of the factors using factors array
   f. Write a method to find the sum of square of the factors using ***Math.pow()*** method

```
package Day4.LabPractice_L2;

import java.util.Scanner;

public class LP1 {

   public static void main(String[] args){

       Scanner input = new Scanner(System.in);

       System.out.print("Enter a number: ");

       int number = input.nextInt();

       int[] factors = findFactors(number);

       System.out.print("Factors of " + number + ": ");

       for (int factor : factors) {

           System.out.print(factor + " ");

       }

       System.out.println("\nSum of Factors: " + sumOfFactors(factors));

       System.out.println("Sum of Squares of Factors: " +
sumOfSquaresOfFactors(factors));

       System.out.println("Product of Factors: " +
productOfFactors(factors));

       input.close();

   }

   public static int[] findFactors(int number) {

       int count = 0;
```

```
        for (int i = 1; i <= number; i++) {
            if (number % i == 0) {
                count++;
            }
        }
        int[] factors = new int[count];
        int index = 0;
        for (int i = 1; i <= number; i++) {
            if (number % i == 0) {
                factors[index++] = i;
            }
        }
        return factors;
    }
    public static int sumOfFactors(int[] factors) {
        int sum = 0;
        for (int factor : factors) {
            sum += factor;
        }
        return sum;
    }
    public static int sumOfSquaresOfFactors(int[] factors) {
        int sum = 0;
        for (int factor : factors) {
            sum += Math.pow(factor, 2);
        }
        return sum;
    }
    public static long productOfFactors(int[] factors) {
        long product = 1;
        for (int factor : factors) {
            product *= factor;
        }
```

```
        return product;
    }
}
```

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.

   **Hint =>**

   a. Take the user input number and check whether it's a Natural number, if not exit
   b. Write a Method to find the sum of n natural numbers using **recursion**
   c. Write a Method to find the sum of n natural numbers using the formulae n*(n+1)/2
   d. Compare the two results and print the result

```
package Day4.LabPractice_L2;
import java.util.Scanner;
public class LP2 {
    public static int sumRecursive(int n) {
        if (n == 1) return 1;
        return n + sumRecursive(n - 1);
    }
    public static int sumFormula(int n) {
        return n * (n + 1) / 2;
    }
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a natural number: ");
        int n = input.nextInt();
        if (n <= 0) {
            System.out.println("Invalid input! Please enter a positive
natural number.");
        } else {
            // Calculate sum using recursion
            int sumUsingRecursion = sumRecursive(n);

            // Calculate sum using formula
            int sumUsingFormula = sumFormula(n);
```

```
            System.out.println("\nResults: \nSum using Recursion: " +
sumUsingRecursion);
            System.out.println("Sum using Formula: " + sumUsingFormula);
            if (sumUsingRecursion == sumUsingFormula) {
                System.out.println("Both computations match!");
            } else {
                System.out.println("Mismatch detected!");
            }
        }
        input.close();
    }
}
```

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

   **Hint =>**

   a. The LeapYear program only works for year >= 1582, corresponding to a year in the Gregorian calendar.
   b. Also Leap year is divisible by 4 and not divisible by 100 or divisible by 400
   c. Write a method to check for Leap Year using the conditions a and b

```
package Day4.LabPractice_L2;
import java.util.Scanner;
public class LP3 {
    public static boolean isLeapYear(int year) {
        if (year < 1582) {
            System.out.println("The year must be 1582 or later (Gregorian
calendar).");
            return false;
        }
        return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
    }
    public static void main(String[] args){
        Scanner input = new Scanner(System.in);
        System.out.print("Enter a year (>=1582): ");
        int year = input.nextInt();
```

```
        if (isLeapYear(year)) {
            System.out.println(year + " is a Leap Year");
        } else {
            System.out.println(year + " is NOT a Leap Year");
        }
        input.close();
    }
}
```

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. E.g.

   ***public static double convertKmToMiles(double km)*** **=>**

   a. Method To convert kilometers to miles and return the value. Use the following code
      `double km2miles = 0.621371;`
   b. Method to convert miles to kilometers and return the value. Use the following code
      `double miles2km = 1.60934;`
   c. Method to convert meters to feet and return the value. Use the following code to convert
      `double meters2feet = 3.28084;`
   d. Method to convert feet to meters and return the value. Use the following code to convert
      `double feet2meters = 0.3048;`

```
package Day4.LabPractice_L2;
import java.util.Scanner;
public class LP4 {
    public static double convertKmToMiles(double km) {
        double km2miles = 0.621371;
        return km * km2miles;
    }
    public static double convertMilesToKm(double miles) {
        double miles2km = 1.60934;
        return miles * miles2km;
    }
    public static double convertMetersToFeet(double meters) {
        double meters2feet = 3.28084;
        return meters * meters2feet;
```

```
    }
    public static double convertFeetToMeters(double feet) {
        double feet2meters = 0.3048;
        return feet * feet2meters;
    }
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter kilometers: ");
        double km = input.nextDouble();
        System.out.print("Enter miles: ");
        double miles = input.nextDouble();
        System.out.print("Enter meters: ");
        double meters = input.nextDouble();
        System.out.print("Enter feet: ");
        double feet = input.nextDouble();
        System.out.println(km + " km = " + convertKmToMiles(km) + "
miles");
        System.out.println(miles + " miles = " + convertMilesToKm(miles) +
" km");
        System.out.println(meters + " meters = " +
convertMetersToFeet(meters) + " feet");
        System.out.println(feet + " feet = " + convertFeetToMeters(feet) +
" meters");
    }
}
```

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. E.g.

   ***public static double convertYardsToFeet(double yards)*** **=>**

   a. Method to convert yards to feet and return the value. Use following code to convert
      `double yards2feet = 3;`
   b. Method to convert feet to yards and return the value. Use following code to convert
      `double feet2yards = 0.333333;`
   c. Method to convert meters to inches and return the value. Use following code to convert
      `double meters2inches = 39.3701;`

d. Method to convert inches to meters and return the value. Use following code to convert
`double inches2meters = 0.0254;`
e. Method to convert inches to centimeters and return the value. Use the following code
`double inches2cm = 2.54;`

```
package Day4.LabPractice_L2;
import java.util.Scanner;
public class LP5 {
    public static double convertYardsToFeet(double yards) {
        double yards2feet = 3;
        return yards * yards2feet;
    }
    public static double convertFeetToYards(double feet) {
        double feet2yards = 0.333333;
        return feet * feet2yards;
    }
    public static double convertMetersToInches(double meters) {
        double meters2inches = 39.3701;
        return meters * meters2inches;
    }
    public static double convertInchesToMeters(double inches) {
        double inches2meters = 0.0254;
        return inches * inches2meters;
    }
    public static double convertInchesToCentimeters(double inches) {
        double inches2cm = 2.54;
        return inches * inches2cm;
    }
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);

        System.out.print("Enter yards: ");
        double yards = input.nextDouble();
        System.out.print("Enter feet: ");
        double feet = input.nextDouble();
```

```
        System.out.print("Enter meters: ");
        double meters = input.nextDouble();
        System.out.print("Enter inches: ");
        double inches = input.nextDouble();
        System.out.print("Enter inches for cm conversion: ");
        double inchesCm = input.nextDouble();
        System.out.println(yards + " yards = " + convertYardsToFeet(yards)
+ " feet");
        System.out.println(feet + " feet = " + convertFeetToYards(feet) +
" yards");
        System.out.println(meters + " meters = " +
convertMetersToInches(meters) + " inches");
        System.out.println(inches + " inches = " +
convertInchesToMeters(inches) + " meters");
        System.out.println(inchesCm + " inches = " +
convertInchesToCentimeters(inchesCm) + " cm");
        input.close();
    }
}
```

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. E.g.

   ***public static double convertFarhenheitToCelsius(double farhenheit)*** **=>**

   a. Method to convert Fahrenheit to Celsius and return the value. Use the following code
      `double farhenheit2celsius = (farhenheit - 32) * 5 / 9;`
   b. Method to convert Celsius to Fahrenheit and return the value. Use the following code
      `double celsius2farhenheit = (celsius * 9 / 5) + 32;`
   c. Method to convert pounds to kilograms and return the value. Use the following code
      `double pounds2kilograms = 0.453592;`
   d. Method to convert kilograms to pounds and return the value. Use the following code
      `double kilograms2pounds = 2.20462;`
   e. Method to convert gallons to liters and return the value. Use following code to convert
      `double gallons2liters = 3.78541;`
   f. Method to convert liters to gallons and return the value. Use following code to convert
      `double liters2gallons = 0.264172;`

```
package Day4.LabPractice_L2;

import java.util.Scanner;
```

```java
public class LP6 {
    public static double convertFahrenheitToCelsius(double fahrenheit) {
        return (fahrenheit - 32) * 5 / 9;
    }
    public static double convertCelsiusToFahrenheit(double celsius) {
        return (celsius * 9 / 5) + 32;
    }
    public static double convertPoundsToKilograms(double pounds) {
        double pounds2kilograms = 0.453592;
        return pounds * pounds2kilograms;
    }
    public static double convertKilogramsToPounds(double kilograms) {
        double kilograms2pounds = 2.20462;
        return kilograms * kilograms2pounds;
    }
    public static double convertGallonsToLiters(double gallons) {
        double gallons2liters = 3.78541;
        return gallons * gallons2liters;
    }
    public static double convertLitersToGallons(double liters) {
        double liters2gallons = 0.264172;
        return liters * liters2gallons;
    }
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Enter Fahrenheit: ");
        double fahrenheit = input.nextDouble();
        System.out.print("Enter Celsius: ");
        double celsius = input.nextDouble();
        System.out.print("Enter Pounds: ");
        double pounds = input.nextDouble();
        System.out.print("Enter Kilograms: ");
        double kilograms = input.nextDouble();
```

```
        System.out.print("Enter Gallons: ");

        double gallons = input.nextDouble();

        System.out.print("Enter Liters: ");

        double liters = input.nextDouble();

        System.out.println(fahrenheit + " Fahrenheit = " +
convertFahrenheitToCelsius(fahrenheit) + " Celsius");

        System.out.println(celsius + " Celsius = " +
convertCelsiusToFahrenheit(celsius) + " Fahrenheit");

        System.out.println(pounds + " Pounds = " +
convertPoundsToKilograms(pounds) + " Kilograms");

        System.out.println(kilograms + " Kilograms = " +
convertKilogramsToPounds(kilograms) + " Pounds");

        System.out.println(gallons + " Gallons = " +
convertGallonsToLiters(gallons) + " Liters");

        System.out.println(liters + " Liters = " +
convertLitersToGallons(liters) + " Gallons");

        input.close();

    }

}
```

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.

   **Hint =>**

   a. Create a class ***public class StudentVoteChecker*** and define a method ***public boolean canStudentVote(int age)*** which takes in age as a parameter and returns true or false
   b. Inside the method firstly validate the age for a negative number, if a negative return is false cannot vote. For valid age check for age is 18 or above return true; else return false;
   c. In the main function define an array of 10 integer elements, loop through the array by take user input for the student's age, call canStudentVote() and display the result

```
package Day4.LabPractice_L2;

import java.util.Scanner;

public class LP7 {

    public static boolean canStudentVote(int age) {

        if (age < 0) return false;
```

```
        return age >= 18;
    }
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int[] ages = new int[10];
        for (int i = 0; i < ages.length; i++) {
            System.out.print("Enter age of student " + (i + 1) + ": ");
            ages[i] = input.nextInt();
        }
        System.out.println();
        for (int i = 0; i < ages.length; i++) {
            if (canStudentVote(ages[i])) {
                System.out.println("Student " + (i + 1) + " (Age: " +
ages[i] + ") can vote.");
            } else {
                System.out.println("Student " + (i + 1) + " (Age: " +
ages[i] + ") cannot vote.");
            }
        }
        input.close();
    }
}
```

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

   **Hint =>**

   a. Take user input for age and height for the 3 friends and store it in two arrays each to store the values for age and height of the 3 friends
   b. Write a Method to find the youngest of the 3 friends
   c. Write a Method to find the tallest of the 3 friends

```
package Day4.LabPractice_L2;
import java.util.Scanner;
public class LP8 {
    public static String findYoungest(String[] names, int[] ages) {
```

```
        int minAge = ages[0];
        StringBuilder youngestFriends = new StringBuilder(names[0]);

        for (int i = 1; i < ages.length; i++) {
            if (ages[i] < minAge) {
                minAge = ages[i];
                youngestFriends = new StringBuilder(names[i]); // Reset if
new min found
            } else if (ages[i] == minAge) {
                youngestFriends.append(", ").append(names[i]); // Append
if tie
            }
        }
        return youngestFriends.toString();
    }

    public static String findTallest(String[] names, double[] heights) {
        double maxHeight = heights[0];
        StringBuilder tallestFriends = new StringBuilder(names[0]);

        for (int i = 1; i < heights.length; i++) {
            if (heights[i] > maxHeight) {
                maxHeight = heights[i];
                tallestFriends = new StringBuilder(names[i]);
            } else if (heights[i] == maxHeight) {
                tallestFriends.append(", ").append(names[i]);
            }
        }
        return tallestFriends.toString();
    }
    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];
        for (int i = 0; i < 3; i++) {
            System.out.print("Enter age of " + names[i] + ": ");
            ages[i] = input.nextInt();
            System.out.print("Enter height (in cm) of " + names[i] + ":
");
            heights[i] = input.nextDouble();
        }
        System.out.println("The youngest friend is: " +
findYoungest(names, ages));
        System.out.println("The tallest friend is: " + findTallest(names,
heights));
        input.close();
    }
}
```

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

   **Hint =>**

   a. Write a Method to Check whether the number is positive or negative
   b. Write a Method to check whether the number is even or odd
   c. Write a Method to compare two numbers and return 1 if number1 > number2 or 0 if both are equal or -1 if number1 < number2
   d. In the main program, Loop through the array using the length call the method ***isPositive()*** and if positive call method ***isEven()*** and print accordingly
   e. If the number is negative, print negative.
   f. Finally compare the first and last element of the array by calling the method ***compare()*** and display if they are equal, greater, or less

```
package Day4.LabPractice_L2;
import java.util.Scanner;
public class LP9 {
    public static boolean isPositive(int number) {
        return number >= 0;
    }
    public static boolean isEven(int number) {
        return number % 2 == 0;
    }
```

```
    public static int compare(int number1, int number2) {
        if (number1 > number2) return 1;
        else if (number1 < number2) return -1;
        else return 0;
    }
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        int[] numbers = new int[5];
        for (int i = 0; i < numbers.length; i++) {
            System.out.print("Enter number " + (i + 1) + ": ");
            numbers[i] = input.nextInt();
        }
        for (int number : numbers) {
            if (isPositive(number)) {
                System.out.print(number + " is Positive and ");
                if (isEven(number)) {
                    System.out.println("Even.");
                } else {
                    System.out.println("Odd.");
                }
            } else {
                System.out.println(number + " is Negative.");
            }
        }
        int result = compare(numbers[0], numbers[numbers.length - 1]);
        if (result == 1) {
            System.out.println(numbers[0] + " > " + numbers[numbers.length
- 1]);
        } else if (result == -1) {
            System.out.println(numbers[0] + " < " + numbers[numbers.length
- 1]);
        } else {
            System.out.println(numbers[0] + " = " + numbers[numbers.length
- 1]);
        }
        input.close();
    }
}
```

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

    **Hint =>**

    a. Take user input in double for the weight (in kg) of the person and height (in cm) for the person and and store it in the corresponding 2D array of 10 rows and 3 columns. The

First Column storing the weight, the second column storing the height in cm and the third column is the BMI

b. Create a Method to find the BMI of every person and populate the array. Use the formula BMI = weight / (height * height). Note unit is kg/m^2. For this convert cm to meter
c. Create a Method to determine the BMI status using the logic shown in the figure below. and return the array of all the persons BMI Status.

| BMI | Status |
|---|---|
| ≤ 18.4 | Underweight |
| 18.5 - 24.9 | Normal |
| 25.0 - 39.9 | Overweight |
| ≥ 40.0 | Obese |

```
package Day4.LabPractice_L2;
import java.util.Scanner;
public class LP10 {
   public static void calculateBMI(double[][] data) {
       for (int i = 0; i < data.length; i++) {
           double weight = data[i][0];
           double heightInMeters = data[i][1] / 100;
           data[i][2] = weight / (heightInMeters * heightInMeters);
       }
   }
   public static String determineBMIStatus(double bmi) {
       if (bmi < 18.5) {
           return "Underweight";
       } else if (bmi >= 18.5 && bmi < 24.9) {
           return "Normal weight";
       } else if (bmi >= 25 && bmi < 29.9) {
           return "Overweight";
       } else {
           return "Obese";
       }
   }
   public static void main(String[] args) {
```

```
        Scanner input = new Scanner(System.in);
        double[][] data = new double[10][3];
        String[] statuses = new String[10];
        for (int i = 0; i < 10; i++) {
            System.out.print("Enter weight (kg) of person " + (i + 1) + ":
");
            data[i][0] = input.nextDouble();
            System.out.print("Enter height (cm) of person " + (i + 1) + ":
");
            data[i][1] = input.nextDouble();
        }
        calculateBMI(data);
        for (int i = 0; i < 10; i++) {
            statuses[i] = determineBMIStatus(data[i][2]);
        }

System.out.println("\nPerson\tWeight(kg)\tHeight(cm)\tBMI\t\tStatus");
        for (int i = 0; i < 10; i++) {
            System.out.printf("%d\t%.2f\t\t%.2f\t\t%.2f\t%s\n", (i + 1),
data[i][0], data[i][1], data[i][2], statuses[i]);
        }
        input.close();
    }
}
```

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

    **Hint =>**

    a. Take a, b, and c as input values to find the roots of x.
    b. The roots are computed using the following formulae

    $delta = b^2 + 4 * a * c$

    If delta is positive the find the two roots using formulae

    $root1\ of\ x = (-b + \sqrt{delta})/(2 * a)$

    $root1\ of\ x = (-b - \sqrt{delta})/(2 * a)$

If delta is zero then there is only one root of x

$root\ of\ x\ =\ \ -\ b/(2\ *\ a)$

If delta is negative return empty array or nothing

c. Write a Method to find find the roots of a quadratic equation and return the roots

```
package Day4.LabPractice_L2;

import java.util.Scanner;

public class LP11 {

   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};

       } else {

           return new double[]{};

       }

   }

   public static void main(String[] args) {

       Scanner input = new Scanner(System.in);

       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();
```

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

        if (roots.length == 2) {

            System.out.println("Two distinct real roots: " + roots[0] + "
and " + roots[1]);

        } else if (roots.length == 1) {

            System.out.println("One real root: " + roots[0]);

        } else {

            System.out.println("No real roots exist.");

        }

        input.close();

    }

}
```

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().

    **Hint =>**

    a. Write a method that generates array of 4 digit random numbers given the size as a parameter as shown in the method signature

    ***public int[] generate4DigitRandomArray(int size)***

    b. Write a method to find average, min and max value of an array

    ***public double[] findAverageMinMax(int[] numbers)***

```
package Day4.LabPractice_L2;

import java.util.Scanner;

import java.util.Arrays;


public class LP12 {

    public static int[] generate4DigitRandomArray(int size) {

        int[] numbers = new int[size];

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

            numbers[i] = 1000 + (int) (Math.random() * 9000);
```

```
        }
        return numbers;
    }
    public static double[] findAverageMinMax(int[] numbers) {
        int min = numbers[0], max = numbers[0], 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);
        System.out.println("Generated Numbers: " +
Arrays.toString(randomNumbers));
        double[] stats = findAverageMinMax(randomNumbers);
        System.out.println("Average: " + stats[0]);
        System.out.println("Minimum: " + (int) stats[1]);
        System.out.println("Maximum: " + (int) stats[2]);
    }
}
```