

QUESTION 1

```
package java_methods;

import java.util.Scanner;

public class MethodsMain {

    public static void main(String[] args) {

        // Create a Scanner object to read user input
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter three numbers
        System.out.println("Enter three numbers:");
        double num1 = scanner.nextDouble();
        double num2 = scanner.nextDouble();
        double num3 = scanner.nextDouble();

        // Call the findSmallest and findLargest methods to determine the smallest and
largest numbers
        double smallest = findSmallest(num1, num2, num3);
        double largest = findLargest(num1, num2, num3);

        // Display the smallest and largest numbers along with a formatted message
        System.out.println("The smallest number: " + smallest);
        System.out.println("The largest number: " + largest);
        System.out.println(smallest + " is your smallest number, and " + largest + " is your
largest number.");

        // Close the Scanner object to release resources
```

```
scanner.close();  
}
```

```
// Method to find the smallest number among three numbers
```

```
public static double findSmallest(double num1, double num2, double num3) {  
    double smallest = num1;  
    if (num2 < smallest) {  
        smallest = num2;  
    }  
    if (num3 < smallest) {  
        smallest = num3;  
    }  
    return smallest;  
}
```

```
// Method to find the largest number among three numbers
```

```
public static double findLargest(double num1, double num2, double num3) {  
    double largest = num1;  
    if (num2 > largest) {  
        largest = num2;  
    }  
    if (num3 > largest) {  
        largest = num3;  
    }  
    return largest;  
}  
}
```

QUESTION 2

```
// JavaProjectExample/src/com/company/Main.java
```

```
package com.company;

import java.util.Scanner;

public class Main {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter marks for Java Programming
        System.out.print("Enter marks for Java Programming: ");
        double javaMarks = scanner.nextDouble();

        // Prompt the user to enter marks for Networking
        System.out.print("Enter marks for Networking: ");
        double networkingMarks = scanner.nextDouble();

        // Prompt the user to enter marks for Maths
        System.out.print("Enter marks for Maths: ");
        double mathsMarks = scanner.nextDouble();

        // Calculate the average marks
        double averageMarks = calculateAverage(javaMarks, networkingMarks,
        mathsMarks);

        // Display individual marks for each unit
        System.out.println("Marks for Java Programming is: " + javaMarks);
        System.out.println("Marks for Networking is: " + networkingMarks);
```

```

        System.out.println("Marks for Maths is: " + mathsMarks);

        // Display the average marks
        System.out.println("The average is: " + averageMarks);
    }

    // Method to calculate the average marks
    public static double calculateAverage(double javaMarks, double networkingMarks,
double mathsMarks) {
        return (javaMarks + networkingMarks + mathsMarks) / 3.0;
    }
}

```

QUESTION 3

Leap year checker

```

import java.util.Scanner;

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

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

        int year = scanner.nextInt();

        if (isLeapYear(year)) {
            System.out.println(year + " is a leap year.");
        } else {
            System.out.println(year + " is not a leap year.");
        }
    }
}

```

```
        scanner.close(); // Closing the Scanner object to free system resources
    }

    // Method to check if a year is a leap year
    public static boolean isLeapYear(int year) {
        // Leap year logic: divisible by 4 but not divisible by 100 unless divisible by 400
        return (year % 4 == 0 && year % 100 != 0) || (year % 400 == 0);
    }
}
```

Triangle area calculator

```
package geometry;

import java.util.Scanner;

public class Triangle {
    private double base;
    private double height;

    // Method to prompt user for base and height input
    public void enterBaseAndHeight() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter the base of the triangle: ");
        base = scanner.nextDouble();

        System.out.print("Enter the height of the triangle: ");
        height = scanner.nextDouble();

        scanner.close(); // Closing the Scanner object after input is read
    }
}
```

```

    }

    // Method to calculate the area of the triangle
    public double calculateArea() {
        return 0.5 * base * height;
    }

    // Method to display the calculated area of the triangle
    public void displayArea() {
        double area = calculateArea();
        System.out.println("The area of the triangle is: " + area);
    }

    public static void main(String[] args) {
        Triangle triangle = new Triangle();
        triangle.enterBaseAndHeight(); // Asking user for input
        triangle.displayArea(); // Displaying the calculated area
    }
}

```

QUESTION 4

```

public class Employee {
    private int id;
    private String name;
    private double salary;

    // Constructor
    public Employee(int id, String name, double salary) {

```

```
this.id = id;

this.name = name;

this.salary = salary;
}

// Non-static method to display employee information
public void displayInfo() {
    System.out.println("Employee ID: " + id);
    System.out.println("Employee Name: " + name);
    System.out.println("Employee Salary: $" + salary);
}

// Static method to calculate bonus based on salary
public static double calculateBonus(double salary) {
    return salary * 0.1; // Assuming 10% bonus
}

// Static method to calculate tax based on salary
public static double calculateTax(double salary) {
    return salary * 0.2; // Assuming 20% tax rate
}

public static void main(String[] args) {
    // Creating employee objects using the constructor
    Employee emp1 = new Employee(1, "John Doe", 50000);
    Employee emp2 = new Employee(2, "Jane Smith", 60000);

    // Calling the non-static method to display employee information
    emp1.displayInfo();
```

```
emp2.displayInfo();
```

```
// Calculating and displaying bonus for emp1 using a static method
```

```
double bonusEmp1 = Employee.calculateBonus(emp1.salary);
```

```
System.out.println("Bonus for " + emp1.name + ": $" + bonusEmp1);
```

```
// Calculating and displaying tax for emp2 using a static method
```

```
double taxEmp2 = Employee.calculateTax(emp2.salary);
```

```
System.out.println("Tax for " + emp2.name + ": $" + taxEmp2);
```

```
}
```

```
}
```