Level 1 Practice Programs

1. Write a program to input the Principal, Rate, and Time values and calculate Simple Interest.

   Hint =>

   a. Simple Interest = Principal * Rate * Time / 100
   b. Take user input for principal, rate, time
   c. Write a method to calculate the simple interest given principle, rate and time as parameters
   d. Output “The Simple Interest is ___ for Principal ___, Rate of Interest ___ and Time ___

```
import java.util.Scanner;


public class SimpleInterestCalculator {


    // Method to calculate simple interest
    public static double calculateSimpleInterest(double principal, double rate, double time) {
        return (principal * rate * time) / 100;
    }


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


        // Taking user input
        System.out.print("Enter Principal amount: ");
        double principal = scanner.nextDouble();


        System.out.print("Enter Rate of Interest: ");
        double rate = scanner.nextDouble();


        System.out.print("Enter Time (in years): ");
        double time = scanner.nextDouble();


        // Calculate simple interest
        double simpleInterest = calculateSimpleInterest(principal, rate, time);
```

```
        // Display output
        System.out.println("The Simple Interest is " + simpleInterest + " for Principal " +
principal +
                        ", Rate of Interest " + rate + " and Time " + time);


        scanner.close();
    }
}
```

```
Enter Principal amount: 1000
Enter Rate of Interest: 5
Enter Time (in years): 1
The Simple Interest is 50.0 for Principal 1000.0, Rate of Interest 5.0 and Time 1.0
```

---

1. Create a program to find the maximum number of handshakes among N number of students.

   Hint =>

   a. Get integer input for number of students
   b. Use the combination = (n * (n - 1)) / 2 formula to calculate the maximum number of possible handshakes.
   c. Write a method to use the combination formulae to calculate the number of handshakes

```
import java.util.Scanner;

public class HandshakeCalculator {

    // Method to calculate maximum number of handshakes
    public static int calculateHandshakes(int n) {
        return (n * (n - 1)) / 2;
    }

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

```
        // Taking user input
        System.out.print("Enter the number of students: ");
        int n = scanner.nextInt();

        // Validate input
        if (n < 2) {
            System.out.println("At least 2 students are required for a handshake.");
        } else {
            // Calculate maximum handshakes
            int maxHandshakes = calculateHandshakes(n);

            // Display output
            System.out.println("The maximum number of handshakes among " + n + " students
is: " + maxHandshakes);
        }

        scanner.close();
    }
}
```

Enter the number of students: 72

The maximum number of handshakes among 72 students is: 2556

---

1. Create a program to find the maximum number of handshakes among N number of students.

   Hint =>

   a. Get integer input for numberOfStudents variable.
   b. Use the combination = (n * (n - 1)) / 2 formula to calculate the maximum number of possible handshakes.
   c. Display the number of possible handshakes.

```
import java.util.Scanner;

public class HandshakeCalculator1 {
```

```
    // Method to calculate maximum number of handshakes
    public static int calculateHandshakes(int numberOfStudents) {
        return (numberOfStudents * (numberOfStudents - 1)) / 2;
    }

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

        // Taking user input
        System.out.print("Enter the number of students: ");
        int numberOfStudents = scanner.nextInt();

        // Validate input
        if (numberOfStudents < 2) {
            System.out.println("At least 2 students are required for a handshake.");
        } else {
            // Calculate maximum handshakes
            int maxHandshakes = calculateHandshakes(numberOfStudents);

            // Display output
            System.out.println("The maximum number of handshakes among " +
numberOfStudents + " students is: " + maxHandshakes);
        }

        scanner.close();
    }
}
```

Enter the number of students: 72

The maximum number of handshakes among 72 students is: 2556

---

1. An athlete runs in a triangular park with sides provided as input by the user in meters. If the athlete wants to complete a 5 km run, then how many rounds must the athlete complete

   Hint =>

   a. Take user input for 3 sides of a triangle
   b. The perimeter of a triangle is the addition of all sides and rounds is distance/perimeter
   c. Write a Method to compute the number of rounds user needs to do to complete 5km run

```
import java.util.Scanner;

public class AthleteRunCalculator {

    // Method to compute number of rounds required
    public static int calculateRounds(double side1, double side2, double side3) {
        double perimeter = side1 + side2 + side3;
        return (int) Math.ceil(5000 / perimeter); // Convert 5km to meters and round up
    }

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

        // Taking user input for three sides of the triangular park
        System.out.print("Enter side 1 of the triangular park (in meters): ");
        double side1 = scanner.nextDouble();

        System.out.print("Enter side 2 of the triangular park (in meters): ");
        double side2 = scanner.nextDouble();

        System.out.print("Enter side 3 of the triangular park (in meters): ");
        double side3 = scanner.nextDouble();

        // Calculate number of rounds needed
        int rounds = calculateRounds(side1, side2, side3);
```

```
        // Display output
        System.out.println("The athlete needs to complete " + rounds + " rounds to finish a 5km run.");

        scanner.close();
    }
}
```

```
Enter side 1 of the triangular park (in meters): 15
Enter side 2 of the triangular park (in meters): 23
Enter side 3 of the triangular park (in meters): 17
The athlete needs to complete 91 rounds to finish a 5km run.
```

---

1. Write a program to check whether a number is positive, negative, or zero.

   Hint => Get integer input from the user. Write a Method to return -1 for negative number, 1 for positive number and 0 if number is zero

```
import java.util.Scanner;

public class NumberCheck {

    // Method to check whether a number is positive, negative, or zero
    public static int checkNumber(int number) {
        if (number > 0) {
            return 1;
        } else if (number < 0) {
            return -1;
        } else {
            return 0;
        }
    }

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

```
        // Taking user input
        System.out.print("Enter a number: ");
        int number = scanner.nextInt();

        // Get result from method
        int result = checkNumber(number);

        // Display output based on result
        if (result == 1) {
            System.out.println("The number is positive.");
        } else if (result == -1) {
            System.out.println("The number is negative.");
        } else {
            System.out.println("The number is zero.");
        }

        scanner.close();
    }
}
```

Enter a number: 23

The number is positive.

---

1. Write a program SpringSeason that takes two int values month and day from the command line and prints “Its a Spring Season” otherwise prints “Not a Spring Season”.

   Hint => Spring Season is from March 20 to June 20. Write a Method to check for Spring season and return a boolean true or false

```
public class SpringSeason {

    // Method to check if the given date is in the Spring season
    public static boolean isSpringSeason(int month, int day) {
        if ((month == 3 && day >= 20) || (month == 4) || (month == 5) || (month == 6 && day <= 20)) {
            return true;
```

```
        }
        return false;
    }

    public static void main(String[] args) {
        // Ensure command line arguments are provided
        if (args.length < 2) {
            System.out.println("Please provide month and day as command-line arguments.");
            return;
        }

        // Parse command line arguments
        int month = Integer.parseInt(args[0]);
        int day = Integer.parseInt(args[1]);

        // Check if it's spring season
        if (isSpringSeason(month, day)) {
            System.out.println("It's a Spring Season");
        } else {
            System.out.println("Not a Spring Season");
        }
    }
}
```

Please provide month and day as command-line arguments.

---

1. Write a program to find the sum of n natural numbers using loop

   Hint => Get integer input from the user. Write a Method to find the sum of n natural numbers using loop

   ```
   import java.util.Scanner;

   public class SumNaturalNumbers {
   ```

```
    // Method to find the sum of n natural numbers
    public static int findSum(int n) {
        int sum = 0;
        for (int i = 1; i <= n; i++) {
            sum += i;
        }
        return sum;
    }

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

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

        // Validate input
        if (n <= 0) {
            System.out.println("Please enter a positive integer.");
        } else {
            // Calculate sum
            int sum = findSum(n);

            // Display output
            System.out.println("The sum of first " + n + " natural numbers is: " + sum);
        }

        scanner.close();
    }
}
```

```
Enter a positive integer: 22
The sum of first 22 natural numbers is: 253
```

---

1. Write a program to find the smallest and the largest of the 3 numbers.

   Hint =>

   a. Take user input for 3 numbers
   b. Write a single method to find the smallest and largest of the three numbers
      public static int[] findSmallestAndLargest(int number1, int number2, int number3)

```
import java.util.Scanner;

public class MinMaxFinder {
    public static void main(String[] args) {
```

```
        Scanner scanner = new Scanner(System.in);

        // Taking input for three numbers
        System.out.print("Enter first number: ");
        int number1 = scanner.nextInt();

        System.out.print("Enter second number: ");
        int number2 = scanner.nextInt();

        System.out.print("Enter third number: ");
        int number3 = scanner.nextInt();

        // Calling the method to find smallest and largest
        int[] result = findSmallestAndLargest(number1, number2, number3);

        // Displaying the results
        System.out.println("Smallest number: " + result[0]);
        System.out.println("Largest number: " + result[1]);

        scanner.close();
    }

    public static int[] findSmallestAndLargest(int number1, int number2, int number3) {
        int smallest = Math.min(number1, Math.min(number2, number3));
        int largest = Math.max(number1, Math.max(number2, number3));

        return new int[]{smallest, largest};
    }
}
```

```
Enter first number: 4
Enter second number: 7
Enter third number: 15
Smallest number: 4
Largest number: 15
```

---

1. Write a program to take 2 numbers and print their quotient and reminder

   Hint =>

   a. Take user input as integer
   b. Use division operator (/) for quotient and moduli operator (%) for reminder
   c. Write Method to find the reminder and the quotient of a number

      public static int[] findRemainderAndQuotient(int number, int divisor)

```
import java.util.Scanner;
```

```
public class MinMaxFinder1 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        // Taking input for two numbers
        System.out.print("Enter the dividend: ");
        int number = scanner.nextInt();

        System.out.print("Enter the divisor: ");
        int divisor = scanner.nextInt();

        // Calling the method to find remainder and quotient
        int[] result = findRemainderAndQuotient(number, divisor);

        // Displaying the results
        System.out.println("Quotient: " + result[0]);
        System.out.println("Remainder: " + result[1]);

        scanner.close();
    }

    public static int[] findRemainderAndQuotient(int number, int divisor) {
        int quotient = number / divisor;
        int remainder = number % divisor;

        return new int[]{quotient, remainder};
    }
}
```

```
Enter the dividend: 30
Enter the divisor: 5
Quotient: 6
Remainder: 0
```

---

1. Create a program to divide N number of chocolates among M children. Print the number of chocolates each child will get and also the remaining chocolates

   Hint =>

   a. Get an integer value from user for the numberOfchocolates and numberOfChildren.
   b. Write the method to find the number of chocolates each child gets and number of remaining chocolates

      public static int[] findRemainderAndQuotient(int number, int divisor)

```
import java.util.Scanner;

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

        // Taking input for number of chocolates and number of children
        System.out.print("Enter the number of chocolates: ");
        int numberOfChocolates = scanner.nextInt();

        System.out.print("Enter the number of children: ");
        int numberOfChildren = scanner.nextInt();

        // Calling the method to find chocolates per child and remaining chocolates
        int[] result = findRemainderAndQuotient(numberOfChocolates, numberOfChildren);

        // Displaying the results
        System.out.println("Each child gets: " + result[0] + " chocolates");
        System.out.println("Remaining chocolates: " + result[1]);

        scanner.close();
    }

    public static int[] findRemainderAndQuotient(int number, int divisor) {
        int quotient = number / divisor;
        int remainder = number % divisor;

        return new int[]{quotient, remainder};
    }
}
```

```
Enter the number of chocolates: 100
Enter the number of children: 42
Each child gets: 2 chocolates
Remaining chocolates: 16
```

---

1. Write a program calculate the wind chill temperature given the temperature and wind speed

   Hint =>

   a. Write a method to calculate the wind chill temperature using the formula

$$windChill = 35.74 + 0.6215 * temp + (0.4275 * temp - 35.75) * windSpeed^{0.16}$$

public double calculateWindChill(double temperature, double windSpeed)

```
import java.util.Scanner;

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

        // Taking input for temperature and wind speed
        System.out.print("Enter the temperature (in Fahrenheit): ");
        double temperature = scanner.nextDouble();

        System.out.print("Enter the wind speed (in mph): ");
        double windSpeed = scanner.nextDouble();

        // Calling the method to calculate wind chill
        double windChill = calculateWindChill(temperature, windSpeed);

        // Displaying the results
        System.out.printf("The wind chill temperature is: %.2f°F\n", windChill);

        scanner.close();
    }

    public static double calculateWindChill(double temperature, double windSpeed) {
        return 35.74 + 0.6215 * temperature + (0.4275 * temperature - 35.75) *
Math.pow(windSpeed, 0.16);
    }
}
```

```
Enter the temperature (in Fahrenheit): 3
Enter the wind speed (in mph): 15
The wind chill temperature is: -15.56°F
```

---

1. Write a program to calculate various trigonometric functions using Math class given an angle in degrees

   Hint =>

   a. Method to calculate various trigonometric functions, Firstly convert to radians and then use Math function to find sine, cosine and tangent.

      public double[] calculateTrigonometricFunctions(double angle)

```
import java.util.Scanner;

public class TrigonometricCalculator {
```

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

        // Taking input for angle in degrees
        System.out.print("Enter the angle in degrees: ");
        double angle = scanner.nextDouble();

        // Calling the method to calculate trigonometric functions
        double[] results = calculateTrigonometricFunctions(angle);

        // Displaying the results
        System.out.printf("Sine: %.4f\n", results[0]);
        System.out.printf("Cosine: %.4f\n", results[1]);
        System.out.printf("Tangent: %.4f\n", results[2]);

        scanner.close();
    }

    public static double[] calculateTrigonometricFunctions(double angle) {
        double radians = Math.toRadians(angle);
        double sine = Math.sin(radians);
        double cosine = Math.cos(radians);
        double tangent = Math.tan(radians);

        return new double[]{sine, cosine, tangent};
    }
}
```

```
Enter the angle in degrees: 90
Sine: 1.0000
Cosine: 0.0000
Tangent: 16331239353195370.0000
```
