1. a) Write a Java program that prompts the user to enter an integer, reads the input, and displays the entered integer on the console.

b) Develop a Java program that reads two floating-point numbers from the user, calculates their average, and displays the result on the console with two decimal places.

```
import java.util.Scanner;

public class ReadInteger {
 public static void main(String[] args) {
 Scanner scanner = new Scanner(System.in);
  System.out.print("Enter an integer: ");
   int enteredInteger = scanner.nextInt();
   System.out.println("You entered: " + enteredInteger);
    scanner.close();
    }
}
```

b.

```
import java.util.Scanner;

public class CalculateAverage {
    public static void main(String[] args) {
        // Create a Scanner object to read input
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter the first floating-point number
        System.out.print("Enter the first floating-point number: ");
        double number1 = scanner.nextDouble();

        // Prompt the user to enter the second floating-point number
        System.out.print("Enter the second floating-point number: ");
        double number2 = scanner.nextDouble();

        // Calculate the average
        double average = (number1 + number2) / 2;

        // Display the result with two decimal places
        System.out.printf("The average is: %.2f%n", average);

        // Close the Scanner to avoid resource leak
        scanner.close();
    }
}
```

2,Implement a Java program that simulates a basic calculator with functionalities to perform addition, subtraction, multiplication, and division.

The program should prompt the user to enter two numbers and an operator (+, -, , /), perform the corresponding operation, and display the result.

Ensure to handle division by zero and invalid operator inputs.

```
import java.util.Scanner;

public class BasicCalculator {
    public static void main(String[] args) {
        // Create a Scanner object to read input
        Scanner scanner = new Scanner(System.in);

        // Prompt the user to enter the first number
        System.out.print("Enter the first number: ");
        double num1 = scanner.nextDouble();

        // Prompt the user to enter the operator
```

```
System.out.print("Enter the operator (+, -, *, /): ");
char operator = scanner.next().charAt(0);

// Prompt the user to enter the second number
System.out.print("Enter the second number: ");
double num2 = scanner.nextDouble();

// Perform the corresponding operation based on the operator
double result = 0;
boolean validOperator = true;

switch (operator) {
    case '+':
        result = num1 + num2;
        break;
    case '-':
        result = num1 - num2;
        break;
    case '*':
        result = num1 * num2;
        break;
    case '/':
        // Check for division by zero
        if (num2 != 0) {
            result = num1 / num2;
        } else {
            System.out.println("Error: Division by zero is not allowed.");
            validOperator = false;
        }
```

```
                break;
            default:
                System.out.println("Error: Invalid operator.");
                validOperator = false;
        }

        // Display the result if the operator is valid
        if (validOperator) {
            System.out.println("Result: " + result);
        }

        // Close the Scanner to avoid resource leak
        scanner.close();
    }
}
```

3. Write an Java program to determine if a number n is happy.

A happy number is a number defined by the following process:

Starting with any positive integer, replace the number by the sum of the squares of its digits. Repeat the process until the number equals 1 (where it will stay), or it loops endlessly in a cycle which does not include 1.Those numbers for which this process ends in 1 are happy.

Print true *if* n *is a happy number, and* false *if not*

```
import java.util.HashSet;
import java.util.Set;

public class HappyNumber {
    public static void main(String[] args) {
        int n = 19; // Replace this with the number you want to check

        boolean isHappy = isHappyNumber(n);
        System.out.println(isHappy);
    }

    public static boolean isHappyNumber(int n) {
        Set<Integer> seen = new HashSet<>();

        while (n != 1 && !seen.contains(n)) {
            seen.add(n);
            n = getNextHappyNumber(n);
        }

        return n == 1;
    }

    private static int getNextHappyNumber(int n) {
        int sum = 0;
        while (n > 0) {
            int digit = n % 10;
            sum += digit * digit;
            n /= 10;
        }
        return sum;
    }
}
```