

DAY-5

Quiz-1

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

a)CODE:

```
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();
    }
}
```

OUTPUT:

```
Enter an integer: 42
You entered: 42
```

b)CODE:

```
import java.util.Scanner;

public class CalculateAverage {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the first floating-point number: ");
        double number1 = scanner.nextDouble();
        System.out.print("Enter the second floating-point number: ");
        double number2 = scanner.nextDouble();
        double average = (number1 + number2) / 2;
        System.out.printf("The average is: %.2f\n", average);
        scanner.close();
    }
}
```

OUTPUT:

```
Enter the first floating-point number: 5.5
Enter the second floating-point number: 3.25
The average is: 4.38
```

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.

CODE:

```
import java.util.Scanner;
public class BasicCalculator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the first number: ");
        double num1 = scanner.nextDouble();
        System.out.print("Enter the operator (+, -, *, /): ");
        char operator = scanner.next().charAt(0);
        System.out.print("Enter the second number: ");
        double num2 = scanner.nextDouble();
        double result = 0;
        switch (operator) {
            case '+':
                result = num1 + num2;
                break;
            case '-':
                result = num1 - num2;
                break;
            case '*':
                result = num1 * num2;
                break;
            case '/':
                if (num2 != 0) {
                    result = num1 / num2;
                } else {
                    System.out.println("Error: Division by zero is not allowed.");
                    System.exit(1);
                }
                break;
            default:
                System.out.println("Error: Invalid operator.");
                System.exit(1);
        }
    }
}
```

```
    }  
    System.out.println("Result: " + result);  
    scanner.close();  
}  
}
```

OUTPUT:

Addition:

```
Enter the first number: 10  
Enter the operator (+, -, *, /): +  
Enter the second number: 5  
Result: 15.0
```

Subtraction:

```
Enter the first number: 8  
Enter the operator (+, -, *, /): -  
Enter the second number: 3  
Result: 5.0
```

Multiplication:

```
Enter the first number: 4.5  
Enter the operator (+, -, *, /): *  
Enter the second number: 2  
Result: 9.0
```

Division:

```
Enter the first number: 12  
Enter the operator (+, -, *, /): /  
Enter the second number: 4  
Result: 3.0
```

Handling division by zero:

```
Enter the first number: 8  
Enter the operator (+, -, *, /): /  
Enter the second number: 0  
ERROR!  
Error: Division by zero is not allowed.
```

Handling invalid operator input:

```
Enter the first number: 5
Enter the operator (+, -, *, /): %
Enter the second number: 3
ERROR!
Error: Invalid operator.
```

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

CODE:

```
import java.util.HashSet;
import java.util.Scanner;
import java.util.Set;
public class HappyNumber {
    public static boolean isHappy(int n) {
        Set<Integer> seen = new HashSet<>();
        while (n != 1 && !seen.contains(n)) {
            seen.add(n);
            n = getNext(n);
        }

        return n == 1;
    }

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

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter a number to check if it's a happy number: ");
        int numberToCheck = scanner.nextInt();
        boolean result = isHappy(numberToCheck);
```

```
        System.out.println("Is " + numberToCheck + " a happy number? " + result);
        scanner.close();
    }
}
```

OUTPUT:

```
Enter a number to check if it's a happy number: 7
Is 7 a happy number? true
```

```
Enter a number to check if it's a happy number: 2
Is 2 a happy number? false
```