**QUIZ 9**

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.

import java.util.Scanner;

public class DisplayEnteredInteger {

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: 27

You entered: 27

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 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: 4.5

Enter the second floating-point number: 4.2

The average is: 4.35

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) {

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;

boolean isValidOperator = true;

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.");

isValidOperator = false;

}

break;

default:

System.out.println("Error: Invalid operator.");

isValidOperator = false;

}

if (isValidOperator) {

System.out.println("Result: " + result);

}

scanner.close();

}

}

OUTPUT:

EXAMPLE 1:

Enter the first number: 10

Enter the operator (+, -, \*, /): +

Enter the second number: 5

Result: 15.0

EXAMPLE 2:

Enter the first number: 8

Enter the operator (+, -, \*, /): \*

Enter the second number: 3

Result: 24.0

EXAMPLE 3:

Enter the first number: 6

Enter the operator (+, -, \*, /): /

Enter the second number: 0

Error: Division by zero is not allowed.

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;

System.out.println(isHappy(n));

}

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;

}

}

OUTPUT:

true