**ASSIGNMENT –30.7.2024**

1)Calculate BMI Using JavaThe user enters his height (in inches) and weight (in pounds). The variables passed by the user are assigned to the float type. After calculating the BMI value, the value will be assigned to the appropriate range and the correct message will appear on the console. You can use the if-else-if ladder for printing the message on the console.

Intervals of BMI index:

16.00 or less = starvation

16.00-16.99 = emaciation

17.00-18.49 = underweight

18.50-22.99 = normal, low range

23.00-24.99 = normal high range

25.00-27.49 = overweight low range

27.50-29.99 = overweight high range

30.00-34.99 = 1st degree obesity

35.00-39.99 = 2nd degree obesity

40.00 or above = 3rd degree obesity

import java.util.Scanner;

public class BMICalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Get user input

System.out.print("Enter your height in inches: ");

float heightInInches = scanner.nextFloat();

System.out.print("Enter your weight in pounds: ");

float weightInPounds = scanner.nextFloat();

// Calculate BMI

float heightInMeters = heightInInches \* 0.0254f;

float weightInKilograms = weightInPounds \* 0.453592f;

float bmi = weightInKilograms / (heightInMeters \* heightInMeters);

// Determine BMI category

String bmiCategory;

if (bmi <= 16.00) {

bmiCategory = "starvation";

} else if (bmi <= 16.99) {

bmiCategory = "emaciation";

} else if (bmi <= 18.49) {

bmiCategory = "underweight";

} else if (bmi <= 22.99) {

bmiCategory = "normal, low range";

} else if (bmi <= 24.99) {

bmiCategory = "normal, high range";

} else if (bmi <= 27.49) {

bmiCategory = "overweight, low range";

} else if (bmi <= 29.99) {

bmiCategory = "overweight, high range";

} else if (bmi <= 34.99) {

bmiCategory = "1st degree obesity";

} else if (bmi <= 39.99) {

bmiCategory = "2nd degree obesity";

} else {

bmiCategory = "3rd degree obesity";

}

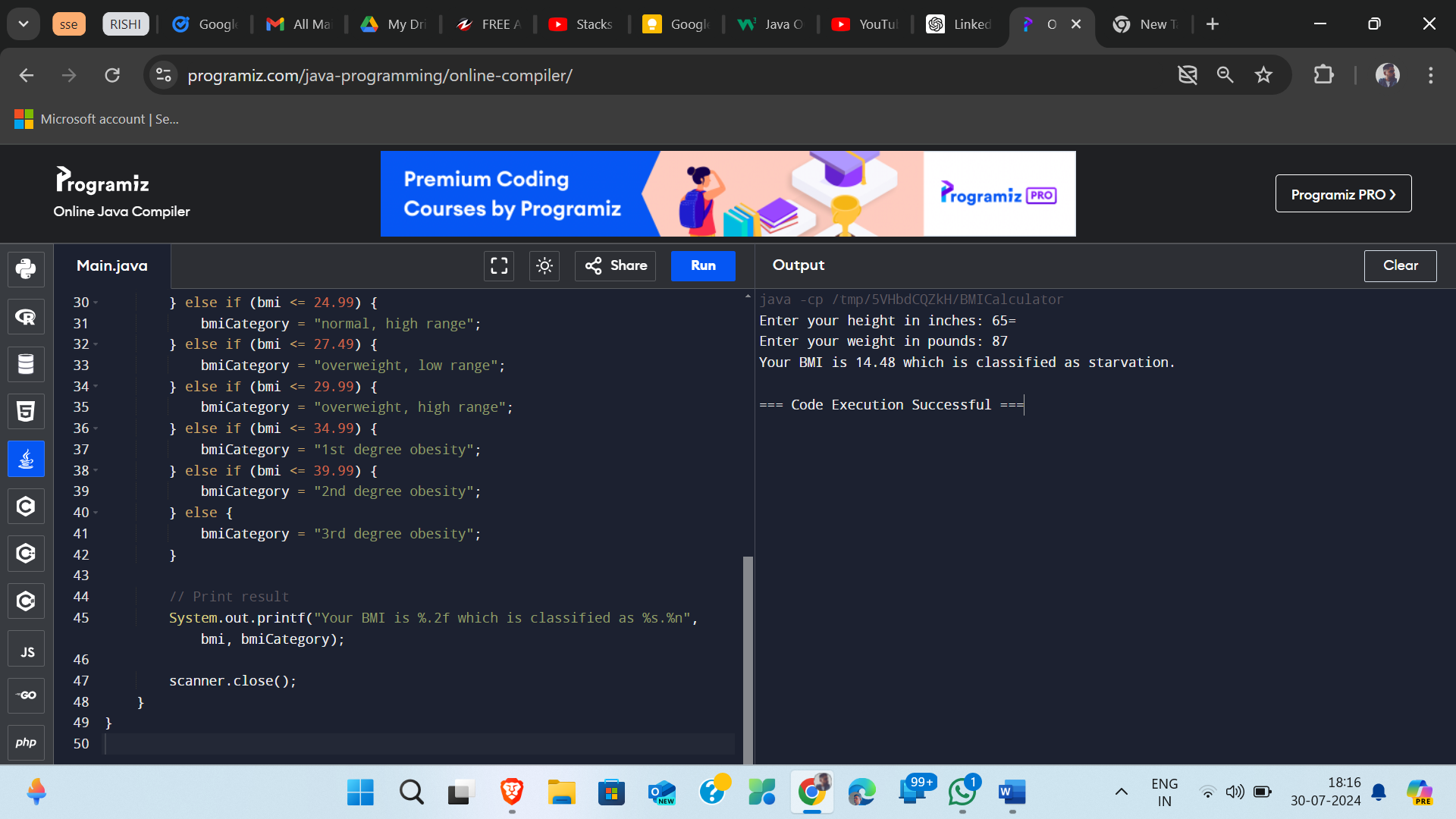
// Print result

System.out.printf("Your BMI is %.2f which is classified as %s.%n", bmi, bmiCategory);

scanner.close();

}

}



2) Write a program that will use the while loop to find the largest and smallest number from the set of 10 randomly drawn integers from 1 to 100. In this task, do not use arrays or other collections.

import java.util.Random;

public class LargestSmallestFinder {

public static void main(String[] args) {

Random random = new Random();

// Initialize the first random number

int number = random.nextInt(100) + 1;

// Set initial values for largest and smallest

int largest = number;

int smallest = number;

// Counter for while loop

int count = 1;

// Print the first number

System.out.println("Random Number " + count + ": " + number);

// Use while loop to generate and compare the remaining 9 numbers

while (count < 10) {

number = random.nextInt(100) + 1;

System.out.println("Random Number " + (count + 1) + ": " + number);

// Update largest and smallest values

if (number > largest) {

largest = number;

}

if (number < smallest) {

smallest = number;

}

// Increment the counter

count++;

}

// Print the results

System.out.println("Largest number: " + largest);

System.out.println("Smallest number: " + smallest);

}

}

