Problem 2: BMICalculator

Exercise Objective: Create an Android app that calculates the Body Mass Index (BMI) and determines the weight category.

Problem Statement 2: The app should have two input fields for weight (in kilograms) and height (in meters). Upon pressing the "Calculate BMI" button, the app should display the BMI value and the corresponding weight category (Underweight, Normal, Overweight, or Obese).

Expected Output: The app displays the BMI value and the corresponding weight category.

activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:padding="16dp"
    tools:context=".MainActivity">

    <!-- Title TextView for BMI Calculator -->
<TextView
android:id="@+id/titleTextView"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="BMI Calculator"
android:textSize="24sp"
android:textStyle="bold"
android:textColor="@android:color/holo_blue_dark"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
android:layout_marginTop="32dp"/>

<EditText
android:id="@+id/etWeight"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginTop="32dp"
android:hint="Weight (kg)"
android:inputType="numberDecimal"
android:minHeight="48dp"
    android:padding="12dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/titleTextView"
    app:layout_constraintWidth_percent="0.9" />
```

```
<EditText
android:id="@+id/etHeight"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginTop="16dp"
android:hint="Height (m)"
android:inputType="numberDecimal"
android:minHeight="48dp"
    android:padding="12dp"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toBottomOf="@+id/etWeight"
    app:layout_constraintWidth_percent="0.9" />

<Button
android:id="@+id/btnCalculateBMI"
android:layout_width="0dp"
android:layout_height="wrap_content"
android:layout_marginTop="24dp"
android:text="Calculate BMI"
android:textSize="18sp"
android:textStyle="bold"
android:backgroundTint="@android:color/holo_green_dark"
android:textColor="@android:color/white"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/etHeight"
app:layout_constraintWidth_percent="0.8" />

<TextView
android:id="@+id/tvBMIResult"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:layout_marginTop="32dp"
android:text="Result: "
android:textSize="20sp"
android:textStyle="italic"
android:textColor="@android:color/black"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/btnCalculateBMI" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

MainActivity.java

```
package com.example.bmicalculator;

import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
```

```
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast; // Import Toast for displaying
messages

import java.text.DecimalFormat; // Import DecimalFormat for
precise formatting if needed, though String.format is often
sufficient

public class MainActivity extends AppCompatActivity {
    // Declare UI elements
    EditText etWeight, etHeight;
    Button btnCalculateBMI;
    TextView tvBMIResult;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // Set the content view to the layout defined in
activity_main.xml
        setContentView(R.layout.activity_main);

        // Initialize UI elements by finding their respective
IDs from the layout
        etWeight = findViewById(R.id.etWeight);
        etHeight = findViewById(R.id.etHeight);
        btnCalculateBMI = findViewById(R.id.btnCalculateBMI);
        tvBMIResult = findViewById(R.id.tvBMIResult);

        // Set an OnClickListener for the Calculate BMI button
        btnCalculateBMI.setOnClickListener(new
View.OnClickListener() {
            @Override
            public void onClick(View v) {
                calculateBmi(); // Call the method to handle
BMI calculation
            }
        });
    }

    /**
     * This method handles the BMI calculation logic,
including input validation
     * and displaying the result.
     */
```

```
    private void calculateBmi() {
        // Get the input strings from the EditText fields
        String weightStr = etWeight.getText().toString();
        String heightStr = etHeight.getText().toString();

        // Check if either input field is empty
        if (weightStr.isEmpty() || heightStr.isEmpty()) {
            Toast.makeText(this, "Please enter both weight and
height.", Toast.LENGTH_SHORT).show();
            tvBMIResult.setText("Result: "); // Clear previous
result
            return; // Exit the method if input is incomplete
        }

        try {
            // Parse the input strings to float values
            float weight = Float.parseFloat(weightStr);
            float height = Float.parseFloat(heightStr);

            // Check if height is zero to prevent division by
zero error
            if (height == 0) {
                Toast.makeText(this, "Height cannot be zero.",
Toast.LENGTH_SHORT).show();
                tvBMIResult.setText("Result: "); // Clear
previous result
                return; // Exit the method
            }

            // Calculate BMI: weight (kg) / (height (m) *
height (m))
            float bmi = weight / (height * height);

            // Get the health assessment based on BMI
            String assessment = getHealthAssessment(bmi);

            // Format and display the BMI result and
assessment
            // Corrected String.format pattern: "%.2f" for
float, "%s" for string
            tvBMIResult.setText(String.format("BMI: %.2f
(%s)", bmi, assessment));

        } catch (NumberFormatException e) {
            // Catch NumberFormatException if the input is not
a valid number
            Toast.makeText(this, "Invalid input. Please enter
```

```
valid numbers for weight and height.",
Toast.LENGTH_LONG).show();
            tvBMIResult.setText("Result: "); // Clear previous
result
            e.printStackTrace(); // Print stack trace for
debugging purposes
        }
    }

    /**
     * Determines the health assessment based on the
calculated BMI.
     * @param bmi The calculated Body Mass Index.
     * @return A string indicating the health category (e.g.,
"Underweight", "Normal").
     */
    private String getHealthAssessment(float bmi) {
        if (bmi < 18.5) {
            return "Underweight";
        } else if (bmi < 25) {
            return "Normal";
        } else if (bmi < 30) {
            return "Overweight";
        } else {
            return "Obese";
        }
    }
}
```