# **Assignment 1** Jan 8, 2025

## Question

Develop a Flutter app using Dart programming that allows users to convert temperatures between Celsius, Fahrenheit, and Kelvin.

The app should include an input field for the temperature value, drop-down menus to select the input and output units, and a button to perform the conversion.

Display the converted temperature result within the app's interface.

## Objective

We are building an application in Flutter that performs conversion among Celsius, Fahrenheit, and Kelvin. It is done in such a way that the user would provide an input and need clicking on ‘Convert’ to perform the conversion.

## Design procedure

WIP.

## Code

import 'package:flutter/material.dart';

void main() => runApp(const MyApp());

class MyApp extends StatelessWidget {

  const MyApp({super.key});

  @override

  Widget build(BuildContext context) {

    return MaterialApp(

      theme: ThemeData(

        colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepOrange),

        useMaterial3: true,

      ),

      home: const HomePage(),

    );

  }

}

class HomePage extends StatefulWidget {

  const HomePage({super.key});

  @override

  State<HomePage> createState() => \_HomePageState();

}

class \_HomePageState extends State<HomePage> {

  // Default units

  String \_unit1 = 'Celsius';

  String \_unit2 = 'Fahrenheit';

  final TextEditingController \_controller1 = TextEditingController();

  final TextEditingController \_controller2 = TextEditingController();

  List<String> units = ['Celsius', 'Fahrenheit', 'Kelvin'];

  void \_convert() {

    double value = double.tryParse(\_controller1.text) ?? 0.0;

    double result = 0.0;

    if (\_unit1 == 'Celsius' && \_unit2 == 'Fahrenheit') {

      result = (value \* 9 / 5) + 32;

    } else if (\_unit1 == 'Celsius' && \_unit2 == 'Kelvin') {

      result = value + 273.15;

    } else if (\_unit1 == 'Fahrenheit' && \_unit2 == 'Celsius') {

      result = (value - 32) \* 5 / 9;

    } else if (\_unit1 == 'Fahrenheit' && \_unit2 == 'Kelvin') {

      result = (value - 32) \* 5 / 9 + 273.15;

    } else if (\_unit1 == 'Kelvin' && \_unit2 == 'Celsius') {

      result = value - 273.15;

    } else if (\_unit1 == 'Kelvin' && \_unit2 == 'Fahrenheit') {

      result = (value - 273.15) \* 9 / 5 + 32;

    } else {

      result = value;

    }

    \_controller2.text = result.toStringAsFixed(2);

  }

  @override

  Widget build(BuildContext context) {

    return Scaffold(

      appBar: AppBar(

        backgroundColor: Theme.of(context).colorScheme.inversePrimary,

        title: const Text('Temperature Converter'),

      ),

      body: Padding(

        padding: const EdgeInsets.all(16.0),

        child: Center(

          child: Column(

            mainAxisAlignment: MainAxisAlignment.center,

            children: [

              \_buildTextField(\_controller1, \_unit1, (String? newValue) {

                setState(() {

                  \_unit1 = newValue!;

                  \_convert();

                });

              }),

              const SizedBox(height: 20),

              \_buildTextField(\_controller2, \_unit2, (String? newValue) {

                setState(() {

                  \_unit2 = newValue!;

                  \_convert();

                });

              }),

            ],

          ),

        ),

      ),

    );

  }

  Widget \_buildTextField(TextEditingController controller, String unit,

      ValueChanged<String?> onChanged) {

    return TextField(

      controller: controller,

      keyboardType: TextInputType.number,

      decoration: InputDecoration(

        border: const OutlineInputBorder(),

        labelText: 'Value',

        suffixIcon: DropdownButton<String>(

          value: unit,

          onChanged: onChanged,

          items: units.map<DropdownMenuItem<String>>((String value) {

            return DropdownMenuItem<String>(

              value: value,

              child: Text(value),

            );

          }).toList(),

          underline: Container(),

        ),

      ),

      onChanged: (\_) => \_convert(),

    );

  }

}

## Output

|  |  |
| --- | --- |
|  |  |
|  |  |
|  |  |

## Discussion

* Ensure that the application handles various edge cases and invalid inputs gracefully.
* Pay attention to the accuracy of the temperature conversion formulas. It’s set to 2 decimal places intentionally.
* Maintain a consistent design language across the application.