









//Task 0: Add your name and Student Id here  
//Name: Daniel Trenholm Student ID: 201202966  
package com.example.quizprogram1  
  
import android.os.Bundle  
import androidx.activity.ComponentActivity  
import androidx.activity.compose.setContent  
import androidx.compose.foundation.layout.Arrangement  
import androidx.compose.foundation.layout.Column  
import androidx.compose.foundation.layout.fillMaxSize  
import androidx.compose.foundation.text.KeyboardOptions  
import androidx.compose.material3.Button  
import androidx.compose.material3.OutlinedTextField  
import androidx.compose.material3.Surface  
import androidx.compose.material3.Text  
import androidx.compose.runtime.Composable  
import androidx.compose.runtime.getValue  
import androidx.compose.runtime.mutableStateOf  
import androidx.compose.runtime.remember  
import androidx.compose.runtime.setValue  
import androidx.compose.ui.Alignment  
import androidx.compose.ui.Modifier  
import androidx.compose.ui.text.TextStyle  
import androidx.compose.ui.text.input.ImeAction  
import androidx.compose.ui.text.input.KeyboardType  
import androidx.compose.ui.unit.sp  
import com.example.quizprogram1.ui.theme.Quizprogram1Theme  
  
class MainActivity : ComponentActivity() {  
 override fun onCreate(savedInstanceState: Bundle?) {  
 super.onCreate(savedInstanceState)  
 *setContent* **{** Quizprogram1Theme **{** Surface(  
 modifier = Modifier.*fillMaxSize*(),  
 ) **{** //Task 2: Define state variables  
 var StringChange by remember**{** *mutableStateOf*("0") **}** //Task 3: Define lambda function for onValueChange of the TextField  
 var onValueChange : (String) -> Unit = **{**StringChange = **it}** //Task 4: Define a lambda function for onClick callback for the "switch" Button  
 var onClick = **{}** //only one i couldn't get  
 //Task 5: replace the call the Converter function with proper parameters  
 Converter(StringChange, false, onValueChange , onClick)  
 **}  
 }  
 }** }  
}  
  
@Composable  
fun Converter(input: String,  
 miles2KM: Boolean,  
 onInputChange: (String) -> Unit,  
 onSwitchButton: ()->Unit) {  
 //Task 6: convert the "input" to double and store it in a variable called "distance" (could be null!)  
 var distance = input.*toDoubleOrNull*()  
 //Task 7: calculate the result, depending on the miles2KM state  
 //note that variable "distance" could be null  
 var result = ""  
 if(miles2KM && distance != null){  
 result = (distance \* 1.60934).toString()  
 } else if(!miles2KM && distance != null) {  
 result = (distance / 1.60934).toString()  
 }  
  
  
 Column (  
 //task 1: align the elements CenterHorizontally  
 horizontalAlignment = Alignment.CenterHorizontally,  
 verticalArrangement = Arrangement.SpaceEvenly  
  
  
 )**{** //Task 0: put your name and student ID in the following Text  
 Text(text = "Daniel Trenholm - 201202966", fontSize = 24.*sp*)  
 //Task 8: change the following to show "Miles to Kilometer" or "Kilometer to Miles" depending on the c2f state  
 if(miles2KM) {  
 Text("Miles to Kilometer", fontSize = 36.*sp*)  
 } else{  
 Text("Kilometers to Miles", fontSize = 36.*sp*)  
 }  
 OutlinedTextField(value = input,  
 label = **{** Text(text = "Input", fontSize = 24.*sp*) **}**,  
 textStyle = TextStyle.Default.copy(fontSize = 24.*sp*),  
 keyboardOptions = KeyboardOptions.Default.copy(  
 keyboardType = KeyboardType.Number,  
 imeAction = ImeAction.Done  
 ),  
 //Task 9: change assignment to the onValueChange function  
 onValueChange = onInputChange  
 )  
  
 Text( text = "Result: " + result.toString(), fontSize = 24.*sp*,  
 )  
 //switch Button  
 Button(  
 //Task 10: replace assignment to the onClick function  
 onClick = onSwitchButton,  
 enabled = true  
 ) **{** Text("switch", fontSize = 24.*sp*)  
 **}  
 }**}  
//Task 11: the input may get lost when you rotate your screen  
//briefly describe a method to help solve this problem.  
//Method  
//You could use the states of the computer, such as onCreate, onPause, onDestroy, etc,  
//to store the values before the state changes, therefore creating a bridge between states  
//and the information that has been input