```java
/* This app is a simple converter for temperature, area, length, and weight
it uses dynamically changes and updates the view and a spinner is used as a drop down menu.
 */
package ca.camosun.androidtemperatureconverter;
import android.support.v7.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Spinner;
import java.util.ArrayList;
public class MainActivity extends AppCompatActivity implements
AdapterView.OnItemSelectedListener {
/* Declared variables for the Lambda functions, for the string when user does not input a
number value.
*/
    private static String nA = "Please enter a number!";
    private ArrayList<ConvertAll> convertAlls;
    private Lambda rightConverter;
    private Lambda leftConverter;
/* scope of the conversion class which will be passed in the lambda functions.
 */
    private class ConvertAll {
        private String labelName;
        private String lLabel;
        private String rLabel;
        private Lambda leftConverter;
        private Lambda rightConverter;
// Takes the variables from the ConvertAll class and controls access to it.
        private ConvertAll(String labelName, String lLabel, String rLabel, Lambda
leftConverter, Lambda rightConverter) {
            this.labelName = labelName;
            this.lLabel = lLabel;
            this.rLabel = rLabel;
            this.leftConverter = leftConverter;
            this.rightConverter = rightConverter;
        }
    }
// Sets up the lambda interface with the type of input allowable
    public interface Lambda {
        double convert(Double input);
    }
/*Called when the activity class is first created and gives information about the layout
resource.
 creates a new array list that contains the Lambda expressions and label which dynamically
change.
 */
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        // Lambda expression which adds the labels and information to the spinner.
        convertAlls = new ArrayList<>();
        convertAlls.add(new ConvertAll("Area", "Ac to Ha", "Ha to AC", (ac) -> ac *
0.404686, (ha) -> ha * 2.47105));
        convertAlls.add(new ConvertAll("Length", "Ft to M", "M to FT", (ft) -> ft * 0.3048,
(m) -> m * 3.28084));
        convertAlls.add(new ConvertAll("Temperature", "F to C", "C to F", (fa) -> (fa -
32.0) * 5.0 / 9.0, (cel) -> cel * 9.0 / 5 + 32));
        convertAlls.add(new ConvertAll("Weight", "Lbs to Kg", "Kg to Lbs", (lbs) -> lbs *
```

```
0.453592, (ki) -> ki * 2.20462));
        // creates an array of spinnerValues to store Area, Length, Temperature, Weight
labels.
        ArrayList<String> spinnerValues = new ArrayList<>();
        for (ConvertAll conversion : convertAlls) {
            spinnerValues.add(conversion.labelName);
        }
        // Creates an adapter for spinner functionality.
        ArrayAdapter<String> adapter;
        adapter = new ArrayAdapter<>(
                this, android.R.layout.simple_spinner_dropdown_item, spinnerValues);
        Spinner spinner = findViewById(R.id.spinner);
        spinner.setAdapter(adapter);
        spinner.setOnItemSelectedListener(this);
        Button leftButton = findViewById(R.id.leftButton);
        leftButton.setText(convertAlls.get(0).lLabel);
        Button rightButton = findViewById(R.id.rightButton);
        rightButton.setText(convertAlls.get(0).rLabel);
    }
    // This is for the left button to dynamically change through the text for the
conversions.
    public void leftButton(View view) {
        EditText converterField = findViewById(R.id.converterField);
        String slField;
        slField = converterField.getText().toString();
        // This is if field is not null then render the text
        try {
            if (leftConverter != null) {

converterField.setText(String.valueOf(leftConverter.convert(Double.parseDouble(slField))));
            }
            // This is is the field is empty while the user clicks on the button it will
let the user know
            //to please input a number.
        } catch (Exception e) {
            converterField.setText(nA);
        }
    }
    // // This is for the right button to dynamically change through the text for the
conversions.
    public void rightButton(View view) {
        EditText converterField;
        converterField = (EditText) findViewById(R.id.converterField);
        String srField = converterField.getText().toString();
        // This is if field is not null then render the text
        try {
            if (rightConverter != null) {

converterField.setText(String.valueOf(rightConverter.convert(Double.parseDouble(srField))));
            }
            // This is is the field is empty while the user clicks on the button it will
let the user know
            //to please input a number.
        } catch (Exception e) {
            converterField.setText(nA);
        }
    }
/*
This is when the adapterView contains no selected item
When the left or right buttons are clicks a message pops up to let
the user know that nothing was selected.
```

```
 */
    public void onNothingSelected(AdapterView<?> parent) {
        Button leftButton = findViewById(R.id.leftButton);
        leftButton.setText(nA);
        Button rightButton = findViewById(R.id.rightButton);
        rightButton.setText(nA);
        leftConverter = null;
        rightConverter = null;
    }
/* This is for the functionality when a button has been clicked
 */
    public void onItemSelected(AdapterView<?> parent, View view, int pos, long id) {
        Button leftButton = findViewById(R.id.leftButton);
        leftButton.setText(convertAlls.get(pos).lLabel);
        Button rightButton = findViewById(R.id.rightButton);
        rightButton.setText(convertAlls.get(pos).rLabel);
        leftConverter = convertAlls.get(pos).leftConverter;
        rightConverter = convertAlls.get(pos).rightConverter;
    }
}
```