```java
package ca.camosun.masterdetailconverter.conversion;

import java.util.ArrayList;

// A Conversion that can be performed.  Contains two buttons to provide conversions between two
// different units of measurement in both directions
public class Conversion {
    // the unique id of the conversion
    private int id;
    // the name to display for the conversion
    private String name;
    // the left button
    private ConversionButton leftButton;
    // the right button
    private ConversionButton rightButton;
    // tracks the next unique id for each new conversion
    private static int IDCOUNTER = 0;

    // returns the next uniqe id for each new conversion and increments static counter
    private int generateID(){
        return IDCOUNTER++;
    }

    // Initializes the Conversion with its name and its two buttons to perfom unit conversions.
    public Conversion(String name, ConversionButton leftButton, ConversionButton rightButton){
        this.id = generateID();
        this.name = name;
        this.leftButton = leftButton;
        this.rightButton = rightButton;
    }

    // public accessor for id
    public int getId(){
        return this.id;
    }

    // Public accessor for name
    public String getName(){
        return this.name;
    }

    // Public accessor for LeftButton
    public ConversionButton getLeftButton(){
        return this.leftButton;
    }

    // Public accessor for RightButton
    public ConversionButton getRightButton(){
        return this.rightButton;
    }

    // Helper method to generate all available conversions.  To add a conversion, simply instantiate
    // a new Conversion object and append to the ArrayList.
    public static ArrayList<Conversion> generateConversions(){

        // Initialize all the conversions
```

```
57          Conversion areaConversion = new Conversion("Area",
58                  new ConversionButton("ha to ac", (Double value) -> {
    return value * 2.471; }),
59                  new ConversionButton("ac to ha", (Double value) -> {
    return value * 0.405; }));
60
61          Conversion tempConversion = new Conversion("Temperature",
62                  new ConversionButton("C to F", (Double value) -> {
    return value * 9.0 / 5.0 + 32.0; }),
63                  new ConversionButton("F to C", (Double value) -> {
    return (value - 32.0) * 5.0 / 9.0; }));
64
65          Conversion lengthConversion = new Conversion("Length",
66                  new ConversionButton("ft to m", (Double value) -> {
    return value * 0.305; }),
67                  new ConversionButton("m to ft", (Double value) -> {
    return value * 3.281; }));
68
69          Conversion weightConversion = new Conversion("Weight",
70                  new ConversionButton("lbs to kg", (Double value) -> {
     return value * 0.454; }),
71                  new ConversionButton("kg to lbs", (Double value) -> {
     return value * 2.205; }));
72
73          // Store all conversions together
74          ArrayList<Conversion> conversions = new ArrayList<>();
75          conversions.add(areaConversion);
76          conversions.add(tempConversion);
77          conversions.add(lengthConversion);
78          conversions.add(weightConversion);
79
80          return conversions;
81      }
82  }
```

```java
package ca.camosun.masterdetailconverter.conversion;

// A button that belongs to a Conversion
public class ConversionButton{
    // the text label to display for the button
    private String name;
    // the lambda expression to call on button click
    private PerformsConversion action;

    // Initializes the ConversionButton with the name and action.  The action must conform to
    // the PerformsConversion interface.  The action is a lambda expression for use later.
    public ConversionButton(String buttonName, PerformsConversion action){
        this.name = buttonName;
        this.action = action;
    }

    // Public accessor for name
    public String getName(){
        return this.name;
    }

    // Public accessor for action
    public PerformsConversion getAction(){
        return this.action;
    }
}

```

```java
package ca.camosun.masterdetailconverter.conversion;

import java.util.HashMap;
import java.util.List;
import java.util.Map;

/**
 *  Contains all Conversions in collections for access throughout the app
 */
public class ConversionContent {

    /**
     * An array of Conversion items.
     */
    public static final List<Conversion> ITEMS = Conversion.generateConversions();

    /**
     * A map of Conversion items, by ID.
     */
    public static final Map<String, Conversion> ITEM_MAP = new HashMap<String, Conversion>();

    /**
     * The number of Conversion items
     */
    private static final int COUNT = ITEMS.size();

    static {
        // Add all Conversion ITEMS to the Map with their ID
        for (int i = 0; i < COUNT; i++) {
            ITEM_MAP.put(Integer.toString(ITEMS.get(i).getId()), ITEMS.get(i));
        }
    }
}

```

```java
package ca.camosun.masterdetailconverter.conversion;

// Interface for all conversion actions to conform to
// All conversion actions must convert an input Double to an output Double
public interface PerformsConversion {
    Double convert(Double value);
}
```