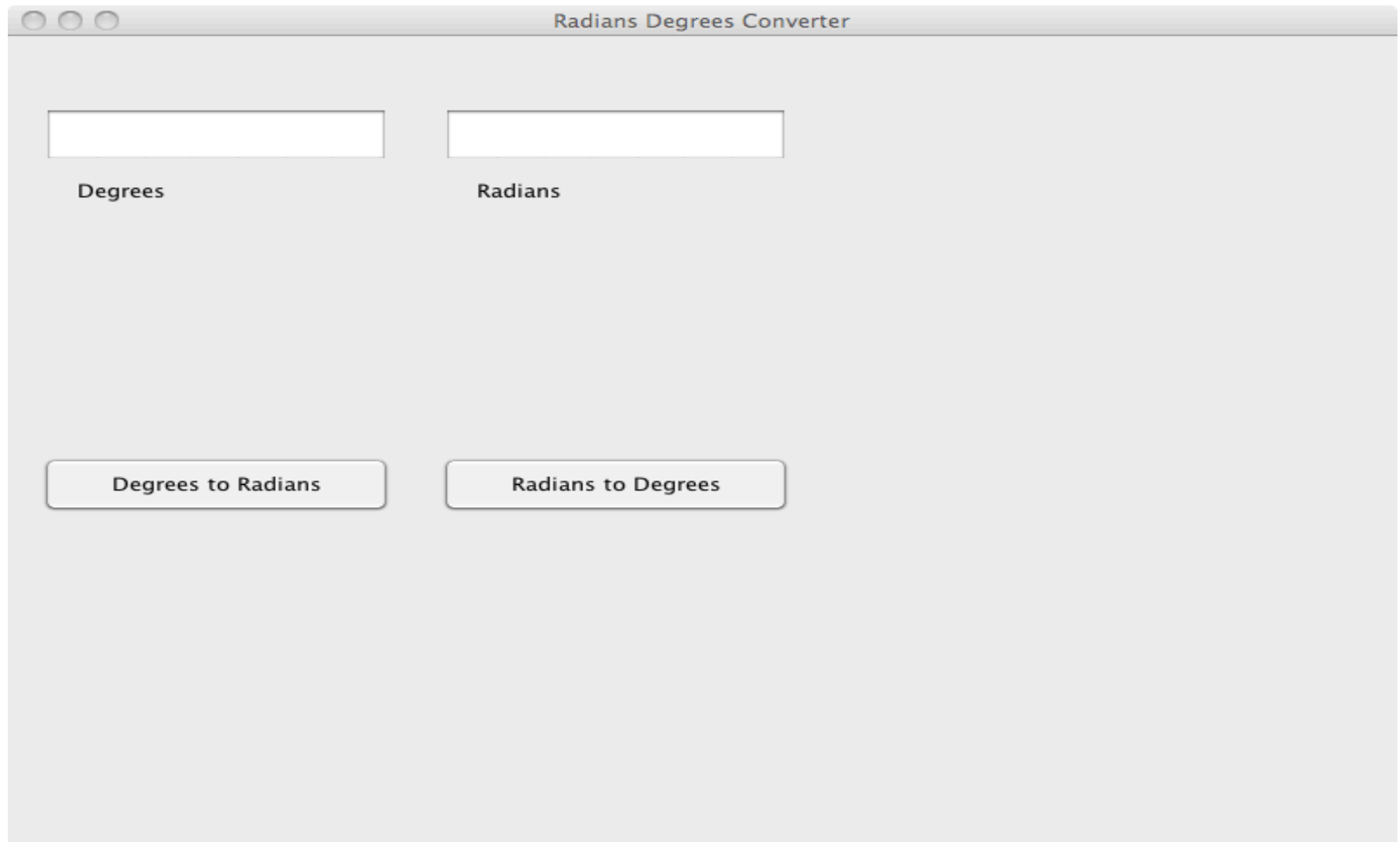


RadiansDegreesConverter



The image shows a screenshot of a graphical user interface for a 'Radians Degrees Converter'. The window has a title bar with three standard macOS window control buttons (red, yellow, green) on the left and the text 'Radians Degrees Converter' in the center. The main content area is light gray and contains two white rectangular input fields. The left input field is labeled 'Degrees' below it, and the right input field is labeled 'Radians' below it. At the bottom of the window, there are two rounded rectangular buttons. The left button is labeled 'Degrees to Radians' and the right button is labeled 'Radians to Degrees'.

Radians Degrees Converter

Degrees

Radians

Degrees to Radians

Radians to Degrees

RadiansDegreesConverter

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.text.*;

public class RadiansDegreesConverter {

    private JFrame window;
    private DegreesToRadiansButton dToR;
    private RadiansToDegreesButton rToD;
    private JTextField degrees;
    private JLabel degreesTitle;
    private JTextField radians;
    private JLabel radiansTitle;
```

RadiansDegreesConverter

```
public RadiansDegreesConverter() {  
    SEE LATER SLIDES  
}
```

```
private String makeDoubleLabel(double f) {  
    DecimalFormat format = new DecimalFormat("#.00000");  
    return format.format(f);  
}
```

```
public double getDegrees() {  
    return new Double(degrees.getText());  
}
```

RadiansDegreesConverter

```
public void setDegrees(double d) {  
    degrees.setText(makeDoubleLabel(d));  
}  
  
public double getRadians() {  
    return new Double(radians.getText());  
}  
  
public void setRadians(double d) {  
    radians.setText(makeDoubleLabel(d));  
}  
  
}
```

RadiansToDegreesButton

```
import javax.swing.*;
```

```
import java.awt.*;
```

```
import java.awt.event.*;
```

```
public class RadiansToDegreesButton extends JButton implements ActionListener {  
    RadiansDegreesConverter controller;
```

```
    public RadiansToDegreesButton(RadiansDegreesConverter c) {  
        setText("Radians to Degrees");  
        controller = c;  
    }
```

```
    public void actionPerformed(ActionEvent e) {  
        double d;  
        double r;  
        r = controller.getRadians();  
        d = r*180.0/Math.PI;  
        controller.setDegrees(d);  
    }
```

DegreesToRadiansButton

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;
```

```
public class DegreesToRadiansButton extends JButton implements ActionListener {  
    RadiansDegreesConverter controller;
```

```
    public DegreesToRadiansButton(RadiansDegreesConverter c) {  
        setText("Degrees to Radians");  
        controller = c;  
    }
```

```
    public void actionPerformed(ActionEvent e) {  
        double d;  
        double r;  
        d = controller.getDegrees();  
        r = d*Math.PI/180.0;  
        controller.setRadians(r);  
    }
```

```
}
```

RadiansDegreesConverter constructor

```
window = new JFrame("Radians Degrees Converter");  
window.setBounds(100, 200, 800, 600);  
window.getContentPane().setLayout(null);  
window.setVisible(true);
```

```
degrees = new JTextField();  
degrees.setBounds(20, 50, 200, 40);  
degrees.setBackground(Color.white);  
window.add(degrees);
```

```
degreesTitle = new JLabel("Degrees");  
degreesTitle.setBounds(40, 90, 200, 40);  
degreesTitle.setForeground(Color.black);  
window.add(degreesTitle);
```

RadiansDegreesConverter constructor (continued)

```
radians = new JTextField();  
radians.setBounds(250, 50, 200, 40);  
radians.setBackground(Color.white);  
window.add(radians);
```

```
radiansTitle = new JLabel("Radians");  
radiansTitle.setBounds(270, 90, 200, 40);  
radiansTitle.setForeground(Color.black);  
window.add(radiansTitle);
```


RadiansDegreesConverter constructor (continued)

```
dToR = new DegreesToRadiansButton(this);  
dToR.setBounds(20, 300, 200, 40);  
dToR.addActionListener(dToR);  
window.add(dToR);
```

```
rToD = new RadiansToDegreesButton(this);  
rToD.setBounds(250, 300, 200, 40);  
rToD.addActionListener(rToD);  
window.add(rToD);
```

```
window.repaint();
```