(1) We will write the code to add a button to the JFrame created by App that when clicked on, creates a new JFrame.

```
public class App {

	public App() {
		javax.swing.JFrame frame = new javax.swing.JFrame("Title");
		javax.swing.JButton button = new javax.swing.JButton("Click Me");



		frame.add(button);
		frame.pack();
		frame.setVisible(true);
		frame.setDefaultCloseOperation(javax.swing.JFrame.EXIT_ON_CLOSE);
	}

	public static void main(String[] args) {
		new App();
	}
}
```

Answer:

```
public class App {

	public App() {
		javax.swing.JFrame frame = new javax.swing.JFrame("Title");
		javax.swing.JButton button = new javax.swing.JButton("Click Me");
		button.addActionListener(new NewJFrameListener());
		frame.add(button);
		frame.pack();
		frame.setVisible(true);
		frame.setDefaultCloseOperation(javax.swing.JFrame.EXIT_ON_CLOSE);
	}

	public static void main(String[] args) {
		new App();
	}
}

public class NewJFrameListener implements java.awt.event.ActionListener {

	public void actionPerformed(java.awt.Event.ActionEvent e) {
		javax.swing.JFrame frame = new javax.swing.JFrame("Another one");
		frame.pack();
		frame.setVisible(true);
	}
}
```

(2) Let's edit the code we have been working on with the moving shape so that there is a new button that appears on the screen with the text Rotate Left on it.

```
public class App {

	private containers.Column _column;
	private Drawing _drawing;

	public App() {
		javax.swing.JFrame frame = new javax.swing.JFrame();
		_column = new containers.Column();
		this.init();
		frame.getContentPane().add(_column);
		frame.pack();
		frame.setVisible(true);
		frame.setDefaultCloseOperation(javax.swing.JFrame.EXIT_ON_CLOSE);
	}

	private void init() {
		_drawing = new Drawing(_column);
		_column.add(new containers.Panel());
		this.setUpButtons();
	}

	private void setUpButtons() {
		containers.Row row = new containers.Row();

		javax.swing.JButton upButton = new javax.swing.JButton("Up");
		upButton.addActionListener(new UpListener(_drawing));
		row.add(upButton);

		javax.swing.JButton downButton = new javax.swing.JButton("Down");
```

```
            downButton.addActionListener(new DownListener(_drawing));
            row.add(downButton);

            javax.swing.JButton leftButton = new javax.swing.JButton("Left");
            leftButton.addActionListener(new LeftListener(_drawing));
            row.add(leftButton);

            javax.swing.JButton rightButton = new javax.swing.JButton("Right");
            rightButton.addActionListener(new RightListener(_drawing));
            row.add(rightButton);

            _column.add(row);









      }

      public static void main(String[] args) {
            new App();
      }
}
```

Write the solution discussed in class below if it differs from your solution.

```
      private void setUpButtons() {
            containers.Row row = new containers.Row();

            javax.swing.JButton upButton = new javax.swing.JButton("Up");
            upButton.addActionListener(new UpListener(_drawing));
            row.add(upButton);

            javax.swing.JButton downButton = new javax.swing.JButton("Down");
            downButton.addActionListener(new DownListener(_drawing));
            row.add(downButton);

            javax.swing.JButton leftButton = new javax.swing.JButton("Left");
            leftButton.addActionListener(new LeftListener(_drawing));
            row.add(leftButton);

            javax.swing.JButton rightButton = new javax.swing.JButton("Right");
            rightButton.addActionListener(new RightListener(_drawing));
            row.add(rightButton);

            _column.add(row);









      }
```

(3) Now edit the Drawing class to appropriate rotate the shape.

```
public class Drawing {

	private graphics.DrawingCanvas _canvas;
	private graphics.Rectangle _shape;

	public Drawing(containers.Column column) {
		_canvas = new graphics.DrawingCanvas();
		_canvas.setColor(new graphics.colors.Black()); //any color we want
		_canvas.setDimension(new java.awt.Dimension(500,500));

		_shape = new graphics.Rectangle();
		_shape.setDimension(new java.awt.Dimension(50,50)); //any dimension
		_shape.setCenterLocation(new java.awt.Point(250, 250)); //any place
		_shape.setColor(new graphics.colors.White()); //any color we want
	}

	public void up() { /* Code removed to save space */ }
	public void down(){ /* Code removed to save space */ }
	public void left(){ /* Code removed to save space */ }
	public void right(){ /* Code removed to save space */ }









}
```

Write the solution discussed in class below if it differs from your solution:

```
public class Drawing {
	private graphics.DrawingCanvas _canvas;
	private graphics.Rectangle _shape;

	public Drawing(containers.Column column) {
		/* Code removed to save space */
	}

	public void up() { /* Code removed to save space */ }
	public void down(){ /* Code removed to save space */ }
	public void left(){ /* Code removed to save space */ }
	public void right(){ /* Code removed to save space */ }









}
```

(4) Write the code for the class that implements the ActionListener interface so that the button we added will actually call the correct method from the Drawing class.

```
public class RotateLeftListener implements java.awt.event.ActionListener {



	public RotateLeftListener() {




	}
	public void actionPerformed(java.awt.event.ActionEvent e) {







	}
}
```

Space to write solution from class if different from what you wrote above:

```
public class RotateLeftListener implements java.awt.event.ActionListener {

	public RotateLeftListener() {




	}
	public void actionPerformed(java.awt.event.ActionEvent e) {




	}
}
```