

Here is a copy of the code that we started with on Monday after the exam.

```
package beginninggraphics;
/**
 * The main (or top-level) object for this program.  It manages and creates the other
 * components that make up the finished program.
 *
 * @author <a href="mailto:adrienne@buffalo.edu">Adrienne M Decker</a>
 */
public class App {

    /**
     * A column to add the other graphical elements of our program to.
     */
    private containers.Column _column;

    /**
     * A drawing object that we will communicate with so that shapes can be drawn.
     */
    private Drawing _drawing;

    /**
     * Creates a new instance of App.  Make sure that you create a JFrame here.
     */
    public App() {
        this.init();
    }

    /**
     * Creates the Drawing object and then gets the buttons set up.  You should not
     * make any changes to this method.
     */
    private void init() {
        _drawing = new Drawing(_column);
        _column.add(new containers.Panel());
        this.setUpButtons();
    }

    /**
     * Sets up the various buttons that the user can interact with on the finished
     * program.  You should not make any changes to this method.
     */
    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);
    }

    /**
     * This method is needed so that this program will run on its own without the aide
     * of the DrJava interactions pane. Right click on this class' name in the package
     * explorer and select Run As -> Java Application to run the program.
     * @param args
     */
    public static void main(String[] args) {
        new App();
    }
}

```

```
package beginninggraphics;
```

```

/**
 * A drawing object. This object consists of a drawing canvas and one square at first.
 * Users can interact with the other components on the same frame as the drawing to add
 * additional elements to the drawing canvas and to manipulate the original square.
 *
 * @author <a href="mailto:adrienne@buffalo.edu">Adrienne M Decker</a>
 *
 * Created on: Feb 17, 2010
 */
public class Drawing {

    /**
     * Create an instance of a drawing canvas and assign it to this instance variable,
     * creating a composition relationship between the Drawing and DrawingCanvas.
     */
    private graphics.DrawingCanvas canvas;

    /**
     * Creates a new instance of Drawing, which means that the drawing canvas and the
     * initial shape are created. The drawing canvas should be added to the column
     * passed in as a parameter to this constructor.
     */
    public Drawing(containers.Column column) {

    }

    /**
     * Move the shape up
     */
    public void up() {

    }

    /**
     * Move the shape down.
     */
    public void down() {

    }

    /**
     * Move the shape to the left.
     */
    public void left() {

    }

    /**
     * Move the shape to the right.
     */
    public void right() {

    }
}

```

(1) Take a moment to underline any parts of this code that you don't understand. Over the course of this week, much will be explained, but if there are still parts that are unclear to you, make sure to ask questions about it. (Or write down some questions here.)

(2) Did the above code compile?

(3) Did the above code run correctly? If not, what happened?

(4) What is a NullPointerException and how are they caused in programs?

(5) How did we fix the NullPointerException? What line of code did we insert?

(6) Then did the program run?

(7) What do we need to add to App to make the program appear on the screen?

```
public class App {  
  
    private containers.Column _column;  
    private Drawing _drawing;  
  
    public App() {  
  
        _column = new containers.Column();  
        this.init();  
  
    }  
  
    private void init() { /*code removed to save space */ }  
    private void setUpButtons() { /*code removed to save space*/ }  
    public static void main(String[] args) { /*code removed to save space*/ }  
}
```

Answer:

```
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() { /*code removed to save space */ }  
    private void setUpButtons() { /*code removed to save space*/ }  
    public static void main(String[] args) { /*code removed to save space*/ }  
}
```

(8) What code do we need to write to get the drawing canvas to appear on the JFrame?

```
public class Drawing {  
    private graphics.DrawingCanvas canvas;  
    public Drawing(containers.Column column) {  
  
    }  
    public void up() {}  
    public void down() {}  
    public void left() {}  
    public void right() {}  
}
```

(9) Now, we need code for a shape that we can move up, down, left and right.

```
public class Drawing {  
    private graphics.DrawingCanvas canvas;  
  
    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));  
  
    }  
    public void up() {}  
    public void down() {}  
    public void left() {}  
    public void right() {}  
}
```

(10) Now, write the code for the up, down, left, right movements of 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() {  
  
    }  
  
    public void down() {  
  
    }  
  
    public void left() {  
  
    }  
  
    public void right() {  
  
    }  
  
}
```