



Monday:
Presidents'
Day



Drawing rectangles for example

1. Define class that extends `JComponent` (or one of its subclasses)
 - Makes the new class a `JComponent` subclass too
2. Implement `paintComponent` method
 - Use `Graphics` object passed to this method
 - Actually better: a `Graphics2D` object since Java 1.2
 - Let that object draw rectangles (or `Rectangle` objects)
 - e.g., [Shapes.java](#) (Fig. 5.26, p. 192)
 - Or [RectangleComponent.java](#) (see links on Slides page)
3. Add the component to a frame for viewing
 - e.g., [ShapesTest](#) or [RectangleViewer.java](#)

Color

- *Current* color applies to text, lines, and fills:

```
g.setColor(Color.RED);
g.drawLine(...); // (or g2.drawLine(...)) draws in red
g.setColor(Color.BLUE);
g.fillRect(...); // (or g2.fillRect(...)) fills with blue
```
- Custom colors available:
 - Can set by float values in range 0.0F to 1.0F:

```
Color gb = new Color(0.0F, 0.7F, 1.0F);
g.setColor(gb);
```
 - Or by int values in range 0 to 255:

```
g.setColor( new Color(0, 255, 175) );
// also shows technique if don't need a reference variable
```
- [ColoredSquareComponent.java](#) and [ColorViewer.java](#)

Drawing more complex shapes

- A simple car, for example – [Car.java](#) (see links)
 - Acts like a `Car` that can draw itself
 - `Car` constructor sets `x` and `y` locations
 - Includes `draw(Graphics2D g2)` method
 - Lets `Graphics2D` object draw lines, ellipses, rectangles
- A class like [CarComponent.java](#) just uses it:

```
Car myCar = new Car(x, y);
myCar.draw(g2); // passes reference to graphics object
```

 - Still need a window to view it, like [CarViewer.java](#)
- Upcoming demo (after chapter 6): animate this drawing

Friday:
2nd Midterm Exam