GUI design using Swing

- The *javax.swing* package is used to create GUI applications.
- The *JFrame* class (part of the swing package) is used to draw windows.
- A basic program generating a *JFrame* would look as follows:

```
import javax.swing.JFrame;
public class OrdinaryFrame {
      public static void main(String[] args){
            int frameWidth = 300;
            int frameHeight = 400;
            String frameTitle = new String("This is a frame.");

            JFrame frame = new JFrame();

            frame.setSize(frameWidth, frameHeight);
            frame.setTitle(frameTitle);
            // Makes sure that when you close the frame,
            // the program stops running
            frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

            // draws the rectangles from RectangleComponent
            TwoRectangleComponents boxes = new TwoRectangleComponents();
            frame.add(boxes);

            frame.setVisible(true);
      }
}
```

- You cannot draw directly on the frame, you must create a component object.
    - The JComponent class is a blank component.
    - To draw on the frame, create a class that extends JComponent:

```
public class TwoRectangleComponents extends JComponent {
      @Override
      public void paintComponent(Graphics g){
            // recover Graphics2D (an extension to the Graphics class)
            Graphics2D g2 = (Graphics2D) g;

            //Construct a rectangle and draw it
            int topLeftX = 5;       // x coordinate of top left corner
            int topLeftY = 10;      // y coordinate of top left corner
            int widthFromTopLeft = 20;
            int heightDownwardFromTopLeft = 30;
            Rectangle box = new Rectangle(topLeftX, topLeftY,
                  widthFromTopLeft, heightDownwardFromTopLeft);
            g2.draw(box);

            // move the box to the right and down, then draw again
            int distanceX = 5;
            int distanceY = 10;
            box.translate(distanceX, distanceY);
            g2.draw(box);
      }
}
```

- Other things to draw:
    - Ellipse2D.Double ellipse = new Ellipse2D.Double(x ,y, width, height);
        - note the x and y are for the top left corner of the bounding box, double is the precision of the variables in which the coordinates, width and height are stored.
    - Line2D.Double segment = new Line2D.Double(x1,y1,x2,y2);
        - A line is drawn connecting the two endpoints--(x1,y1) and (x2,y2)
        - You can also specify two separate points and connect them in a similar fashion:
            - Point2D.Double one = new Point2D.Double(x1,y1)
            - Point2D.Double two = new Point2D.Double(x2,y)
            - Line2D.Double lineSegment = new Line2D.Double(one,end)
    - GraphicsObject.drawString(“Test Message”, x, y); //as opposed to GraphicsObject.draw(ShapeObject);
        - Prints the string with it’s bottom left (or basepoint) corner at the point (x,y)
        - Font type and size is specified in a font object that has parameters for font face, font weight, and font size.
- Setting color:
    - Create a new Color object
        - Color Magenta = new Color(255,0,255);
        - The parameters for this color constructor are values for red, green, and blue respectively, with 0 implying no saturation of that color, and 255 being full saturation.
    - GraphicsObject.setColor(Color.RED);
        - Color.RED refers to one of many constants that Java offers for easy and quick implementations of many common colors.
    - To fill an object with color rather than drawing it in that color, use the GraphicsObject.fill(ShapeObject); method instead of the draw method.
- Applets
    - Applets allow programmers to implement both a graphics component and a graphics viewer in one self-contained class
    - Applets can also run inside of web pages (by implementing a compiled .class file)