The introduction to Java’s graphics context, found in the previous topic *Graphics Context*, uses a Java applet to illustrate the concept. It is also possible to do graphics in a Java application. To create a visible window for a Java application you build an object of class `javax.swing.JFrame`. This class is also a subclass of `java.awt.Container` and, therefore, inherits the `paint` method.

To create an application with a GUI window, make the application a subclass of `javax.swing.JFrame`. The application object will have `paint` as an instance method, which you can morph similarly to an applet. For a more complete discussion of the `JFrame` class and its use to create a GUI window, see the topic *GUI Applications*.

**Example**
The application shown below creates this GUI window. The morphed `paint` method is identical to that in the applet `GraphicsAppletDemo` of the previous section *Graphics Context*. 

```
Welcome to Graphics
```
import javax.swing.JFrame;
import java.awt.*;  // Graphics, Color, Font

public class GraphicsAppDemo extends JFrame
{
    public static void main( String [] args )
    {
        // call app constructor
        new GraphicsAppDemo( "My Window" );
    }

    public GraphicsAppDemo( String title ) // app constructor
    {
        super( title ); // call JFrame constructor
        this.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
        this.setSize( 500, 250 );
        this.setVisible( true );
    }

    public void paint( Graphics g )
    {
        super.paint( g );
        g.setColor( Color.RED );
        g.setFont( new Font("Trebuchet MS", Font.PLAIN, 24) );
        g.drawString( "Welcome to Graphics", 20, 100 );
    }
}

Exercises

1. Modify GraphicsAppletDemo so that it uses a JOptionPane input dialog to read the user’s name and displays a message with the name, such as shown to the right.