```
import java.awt.*;
import java.applet.*;

// This applet computes the value of Pi using a Monte Carlo approach
public class ComputePi extends Applet
{
	private Panel display_panel;
	private Button compute_button;
	private Panel field_panel1;
	private Panel field_panel2;
	private TextField tf_1;
	private TextField tf_2;

	public void init()
	{
		// Build the display panel
		display_panel = new Panel(new BorderLayout());

		// Build the field panel
		PanelFormat pf = new PanelFormat();
		field_panel1 = new Panel();
		pf.initializePanel(field_panel1);
		pf.Subtitle(field_panel1,"Compute Pi");
		pf.Space(field_panel1);
		tf_1 = new TextField(10);
		tf_2 = new TextField(10);
		pf.LabelField(field_panel1,new Label("Number of Monte Carlo Samples"),tf_1);

		field_panel2 = new Panel();
		compute_button = new Button("Compute Pi");
		field_panel2.add(compute_button);
		pf.LabelField(field_panel2,new Label("Value of Pi"),tf_2);

		add("Center",field_panel1);
		add("South",field_panel2);

		// Put in default data
		Integer z = new Integer(10000);
		tf_1.setText(z.toString());
	}

	// Catch events
	//---------------------------------------------------------------------------
	public boolean action(Event event,Object obj)
	{
		Object target = event.target;

		if(target instanceof Button)
		{
			Button button = (Button)target;

			if(target == compute_button)
			{
				// Get inputs
				int n = 0;
				Integer q = null;
```

```
                                boolean f = true;
                                try{q = Integer.valueOf(tf_1.getText()); }
                                catch(NumberFormatException e) {f = false;}
                                if(f)
                                        n = q.intValue();

                                int in_circle = 0;
                                double x;
                                double y;
                                double d;
                                for(int i = 0; i < n; i++)  {

                                        // Get a random number between 0 and 1 for x axis
                                        x = Math.random();

                                        // Get a random number between 0 and 1 for y axis
                                        y = Math.random();

                                        // Figure out distance from origin
                                        // square roots are time consuming, no need to do it
                                        //d = Math.sqrt(x*x + y*y);
                                        d = x*x + y*y;

                                        // Is the point in the circle?
                                        if(d <= 1.0)
                                                in_circle++;
                                }

                                // Compute Pi
                                double p = 4.0*(double)in_circle/(double)n;

                                // Display result
                                Double zd = new Double(p);
                                tf_2.setText(zd.toString());
                        }
                }
                return(true);
        }
}
```