**CS 342 Report for (Lab 01)**

**Code**

// Programmer: Bradley Golden

// Assignment: Lab 1

// Date: September 2, 2015

// Description: This program draws a rectangle and text and let's a

// user have some say in how it looks.

import java.awt.FlowLayout;

import java.awt.Graphics;

import java.awt.event.\*;

import javax.swing.\*;

public class Lab1 extends JApplet implements ActionListener

{

private int centerX; // Center x-coordinate of rectangle to draw

private int centerY; // Center y-coordinate of rectangle to draw

private int appletWidth; //Width of the applet window

private int appletHeight; //Height of the applet window

private int inputWidthInt; // Same as inputWidth but an int

private String inputWidth; // Border width of rectangle to draw, specified by the user

private String inputMessage; // Message to be displayed in rectangle, specified by the user

@Override

public void init() // Set up GUI, initialize variables

{

appletWidth = getWidth(); // Set the width of the applet

appletHeight = getHeight(); // Set the height of the applet

centerX = appletWidth/2; //center of the applet window

centerY = appletHeight/2; //center of the applet window

setLayout(new FlowLayout());

inputMessage = JOptionPane.showInputDialog("Enter a message:"); // Get input message from

// user via dialogue box

inputMessage = inputMessage + " (Bradley Golden)"; // Concatenate input message with my name

do

{

inputWidth = JOptionPane.showInputDialog("Enter the border width:"); // Get width

// from user

inputWidthInt = Integer.parseInt(inputWidth); // Convert string width to int width

} while(inputWidthInt > (appletHeight/2) || inputWidthInt < 0); // appletHeight is used

// because it's smaller

// than applet width

} // end init

@Override

public void paint(Graphics g) // Display results

{

int startX = inputWidthInt; // Upper left x of bounding rect.

int startY = inputWidthInt; // Upper left y of bounding rect.

int borderWidth = appletWidth - inputWidthInt\*2; //width whitespace border

int borderHeight = appletHeight - inputWidthInt\*2; //height of whitespace border

super.paint(g);

g.drawRect(startX, startY, borderWidth, borderHeight); // Draw rectangle

g.drawString(inputMessage, startX+5, centerY); //Draw user input, offset center

// slightly from border

} // end paint

public void actionPerformed(ActionEvent e) // Process user actions

{

repaint(); // Redraw rectangle and text

} // end actionPerformed

} // end Lab1 class

**User Interface**

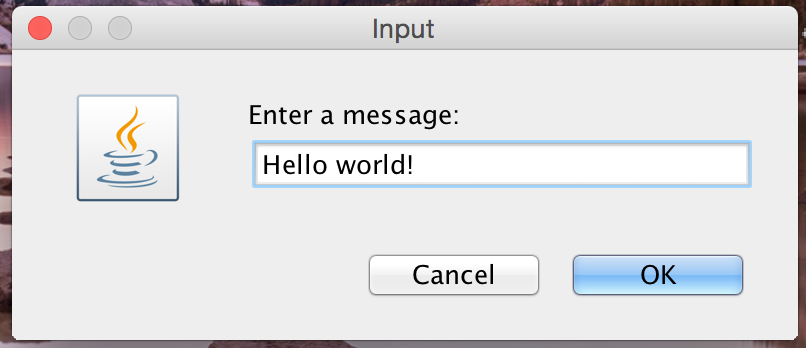
<html>

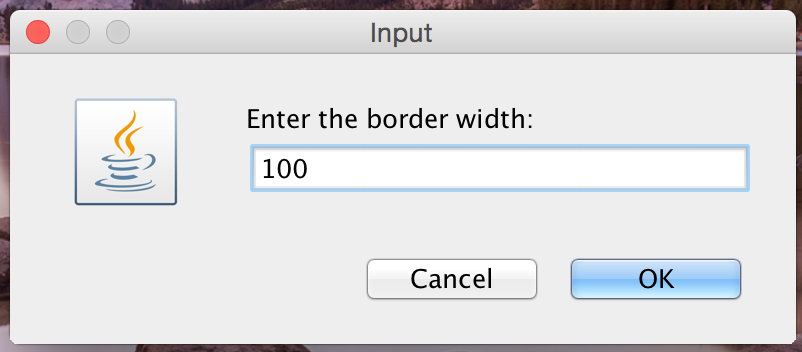
<applet code="Lab1.class" width="400" height="300"></applet>

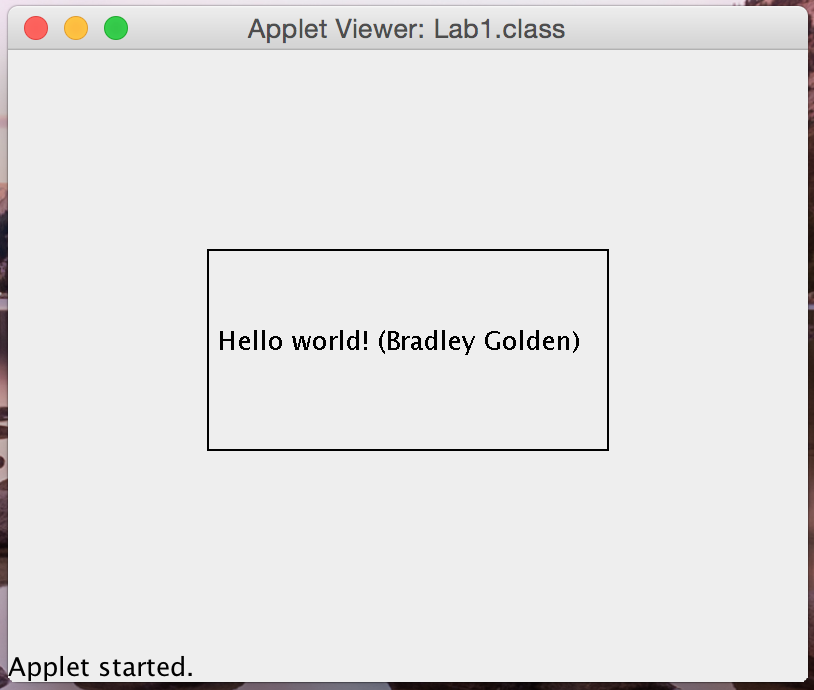
</html>

**Sample Runs**

$ appletviewer lab01.html







**Discussion**

In this lab, I built a GUI program that draws a rectangle and text, allows the user have some say in how it looks, and lets the user specify the size of the rectangle and what text to display. This lab was useful for learning a few concepts including the IPO model and creating a basic Java applet. I also learned basics of how an event listener (Action Event) works.

In particular, there a few difficulties I came across and a few bugs worth mentioning. First, it was a little challenging to determine how to properly align the rectangle within the center of the Applet View. Second, the user input is not checked all that well. In future applications, I would ensure the user enters the correct type when prompted. Third, the text displayed within the rectangle is not formatted to fit perfectly, resulting in runoff if the user’s message is too long.