1. (2 pts) The **FlowLayout** layout manager will arrange components on a container from left to right in rows, and the **GridLayout** layout manager will arrange components on a container in rows and columns.

2. (2 pts) Components in the java.awt package are called **heavyweight** components because they rely heavily on the underlying platform, but most of the components in the javax.swing package are **lightweight** components because they do not rely heavily on their platform.

3. (2 pts) What will be displayed by the following program?

```java
import javax.swing.*;
public class Question3 extends JFrame {
    public Question3(String title) {
        super(title);
    }
    public static void main(String[] args) {
        Question3 question3 = new Question3("Test");
        question3.setSize(400,400);
    }
}
```

**Nothing. Frames are not visible by default.**

4. (2 pts) Recalling that the width and height of a container can be found by the methods getWidth() and getHeight(), and assuming you have at your disposal a variable named g of type Graphics, how would you draw a square of width 50 centered on the center of the container, have a circle filled in blue inside the square and lines drawn in black from opposite corners of the square? You may assume that your code is inside the paint method in the container definition.

```java
    g.drawRect(getWidth()/2-25,getHeight()/2-25,50,50);
    g.setColor(Color.blue);
    g.fillOval(getWidth()/2-25,getHeight()/2-25,50,50);
    g.setColor(Color.black);
    g.drawLine(getWidth()/2-25,getHeight()/2-25,getWidth()/2+25,getHeight()/2+25);
    g.drawLine(getWidth()/2-25,getHeight()/2+25,getWidth()/2+25,getHeight()/2-25);
```

5. (2 pts) Given a Graphics object g, and assuming you are in the paint method in the container definition, how would you draw the figure provided? You may assume the bottom of the outer triangle is the bottom of the container and the top of the outer triangle is the top of the container.

```java
    int[] xpoints = {getWidth(),0,getWidth()/2};
    int[] ypoints = {getHeight(),getHeight(),30};
    g.setColor(Color.red);
    g.fillPolygon(xpoints,ypoints,3);
    xpoints = new int[]{getWidth()-10,10,getWidth()/2};
    ypoints = new int[]{getHeight()-10,getHeight()-10,40};
```
g.setColor(Color.white);
g.setFillPolygon(xpoints, ypoints, 3);