Objectives:

* To describe Java coordinate systems in a GUI component
* To draw things using the methods in the **Graphics** class
* To override the **paintComponent** method to draw things on a GUI component
* To use a panel as a canvas to draw things
* There are 2 exercises please print screen after each exercise

If you need paintComponent to execute (i.e., if you want to update the graphics drawn on a Swing component), you can call method**repaint**, which is inherited by all JComponents indirectly from class Component (package java.awt). The header for repaint is

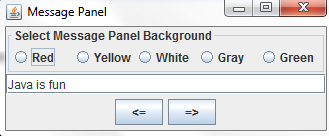
|  |
| --- |
| public void repaint() |

In order to draw things on a component, you need to define a class that extends **JPanel** and overrides its **paintComponent** method to specify what to draw.

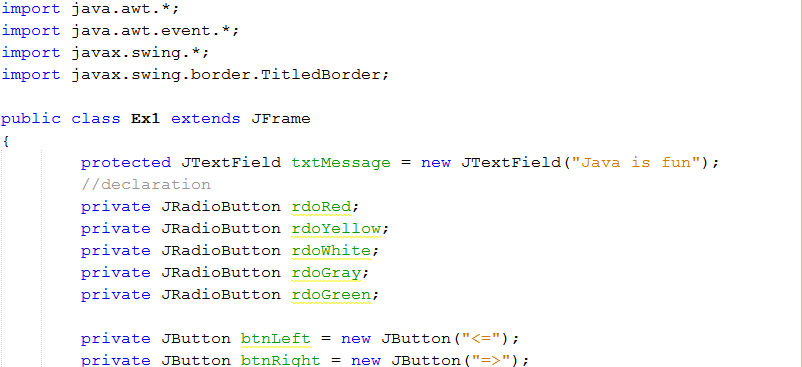
The **paintComponent** method is automatically invoked to paint graphics when the component is first displayed or whenever the component needs to be redisplayed.

Invoking **super.paintComponent(g)** invokes the **paintComponent** method defined in the superclass. This is necessary to ensure that the viewing area is cleared before a new drawing is displayed. Invokes the **drawLine** method to draw a line from (**0**, **0**) to (**50**, **50**). Invokes the **drawString** method to draw a string.

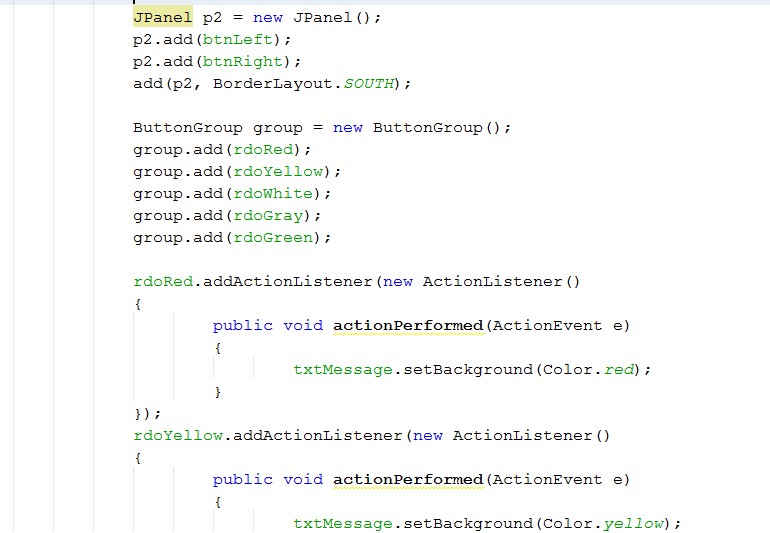
Example #1

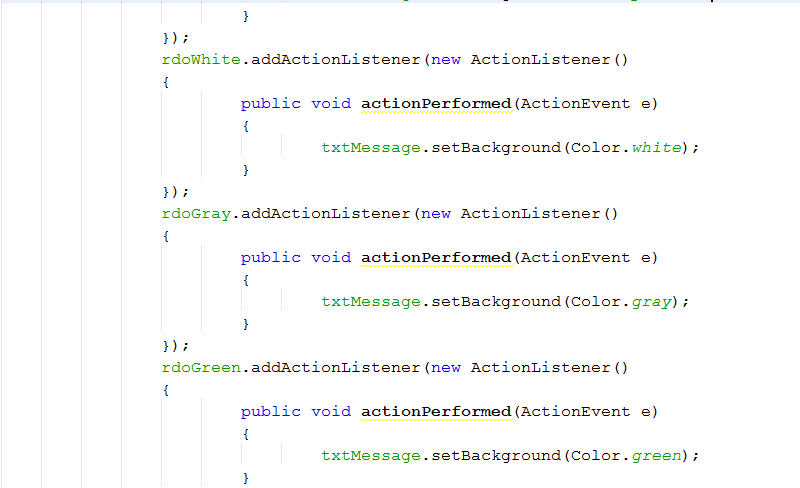


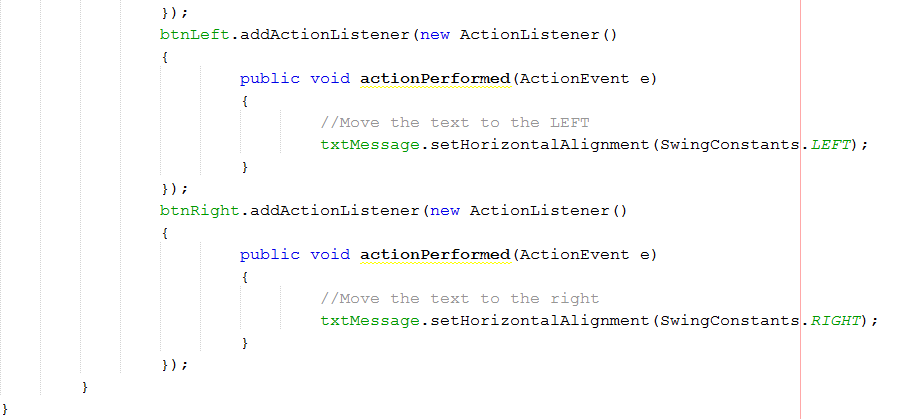
Create a new class and name it **Ex1** using either net beans or eclipse



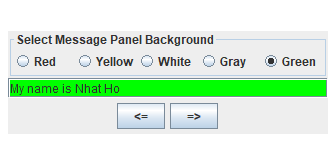
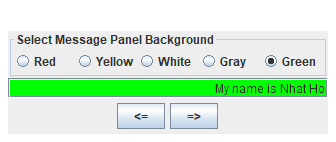




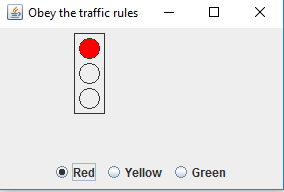




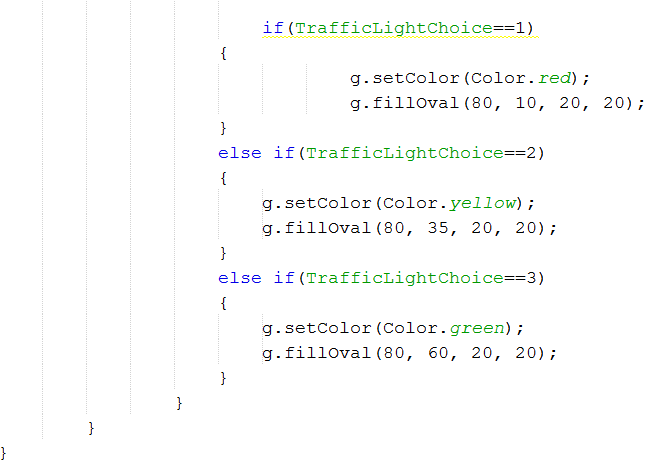
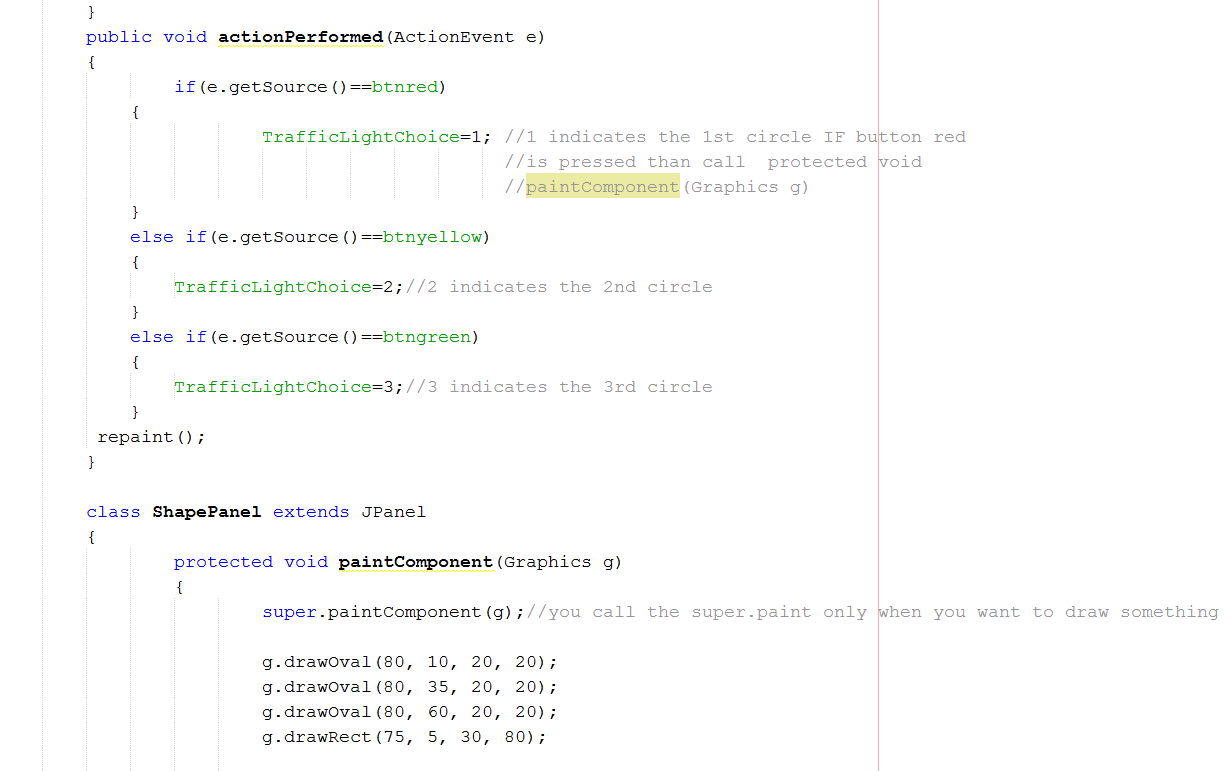
Print screen your running application below here

Example #2

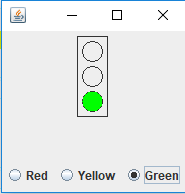
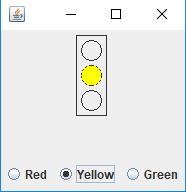
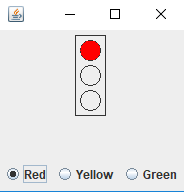






Run the program and notice when you click on the radio buttons it does not respond, create the Action Listeners for each radio button controls

Print screen your running application below here for each light



Submit this document to week 15 Homework