**Name:**

**CST 183 Lab Assignment #9 (Chapter 12)**

**GUI Applications**

**Lab Objectives**

Be able to create a closeable window

Be able to create panels containing buttons

Be able to use different layouts

Be able to handle button events

**Introduction**

In this lab, we will be creating a graphical user interface (GUI) to allow the user to select a button that will change the color of the center panel and radio buttons that will change the color of the text in the center panel. We will need to use a variety of Swing components to accomplish this task.

We will build two panels, a top panel containing three buttons and a bottom panel containing three radio buttons. Layouts will be used to add these panels to the window in the desired positions. A label with instructions will also be added to the window. Listeners will be employed to handle the events desired by the user.

Our final GUI should look like the following



**Task #1 Creating a GUI**

1. Import the required Java libraries.
2. Create a class called ColorFactory that inherits from JFrame.
3. Create named constants for a width of 500 and height of 300 for the frame.
4. Write a default constructor that does the following
	1. Set the title of the window to Color Factory.
	2. Set the size of the window using the constants.
	3. Specify what happens when the close button is clicked.
	4. Get the content pane of the JFrame and set the layout manager to border layout.
	5. Call the method to build the top panel (to be written as directed below).
	6. Add the panel to the north part of the content pane.
	7. Call the method to build the bottom panel (to be written as directed below).
	8. Add this panel to the south part of the content pane.
	9. Create a label that contains the message “Top buttons change the panel color and bottom radio buttons change the text color.”
	10. Add this label to the center part of the content pane.

**Task #2 Writing Private Methods**

1. Write a private method that builds the top panel as follows
	1. Create a panel that contains three buttons, red, orange, and yellow.
	2. Use flow layout for the panel and set the background to be white.
	3. The buttons should be labeled with the color name and also appear in that color.
	4. Set the action command of each button to be the first letter of the color name.
	5. Add button listener that implements action listener for each button.
2. Create a bottom panel in the same way as the top panel above, but use radio buttons with the colors green, blue, and cyan.

**Task #3 Writing Inner Classes**

1. Write a private inner class called ButtonListener that implements ActionListener. It should contain an actionPerformed method to handle the button events. This event handler will handle all button events, so you must get the action command of the event and write a decision structure to determine which color to set the background of the content pane.
2. Write another private inner class called RadioButtonListener, similar to Button listener. It will handle all radio button events, so you will need to check the source of the event and write a decision structure to determine which color should be used for the text of the message.

**Task #4 Running the GUI Program**

1. Write a main method that declares and creates one instance of a ColorFactory, then use the setVisible method to show it on the screen.

**Program Skeleton:**

//import the needed packages
import java.awt.\*;
import java.awt.event.\*;
import javax.swing.\*;

public class ColorFactory extends JFrame
{
 //constants for window size

 //objects needed for GUI
 private JLabel message;
 private Container contentPane;
 private JPanel topPanel;
 private JPanel bottomPanel;
 private JButton redButton;
 private JButton yellowButton;
 private JButton orangeButton;
 private JRadioButton greenButton;
 private JRadioButton blueButton;
 private JRadioButton cyanButton;
 private ButtonGroup radioButtonGroup;

 /\*\*constructor\*/
 public ColorFactory()
 {
 //create a closeable JFrame with a specific size
 setTitle("Color Factory");
 setSize(WIDTH, HEIGHT);
 setDefaultCloseOperation (JFrame.EXIT\_ON\_CLOSE);

 //get the contentPane and set the layout of the window

 //build a panel and add it to the top of the window

 //build a panel and add it to the bottom of the window

 //create a message and add it to the middle of the window
 message = new JLabel("put your message here");
 contentPane.add(message, BorderLayout.CENTER);
 }

 /\*\*create a panel to add to the window\*/
 private void buildTopPanel()
 {
 //create a panel, set the color and layout
 topPanel = new JPanel();
 topPanel.setBackground (Color.white);
 topPanel.setLayout(new FlowLayout());

 //create 3 buttons - red, orange, yellow,

 //setting background, actions commands
 //and adding listener.
 redButton = new JButton("Red");

 orangeButton = new JButton("Orange");

 yellowButton = new JButton("Yellow");

 //add the 3 buttons to the panel

 }

 /\*\*create a panel to add to the window\*/
 private void buildBottomPanel()
 {
 //create a panel, set the color and layout

 //create 3 radio buttons, setting foreground

 //create a radio button group and add the 3
 //radio buttons

 //add listeners to the radio buttons

 //add the radio buttons to the panel
 bottomPanel.add(greenButton);
 bottomPanel.add(blueButton);
 bottomPanel.add(cyanButton);
 }

 /\*\*button listener changes color of background depending
 on which button is clicked\*/
 private class ButtonListener implements ActionListener
 {
 public void actionPerformed (ActionEvent event)
 {
 String whichButton= event.getActionCommand();
 if(whichButton.equals("R"))
 {
 contentPane.setBackground(Color.red);
 }
 else if (whichButton.equals("O"))
 {
 contentPane.setBackground(Color.orange);
 }
 else if (whichButton.equals("Y"))
 {
 contentPane.setBackground(Color.yellow);
 }
 }
 }

 /\*\*radio button listener changes color of foreground (text)
 depending on which button is clicked\*/
 private class RadioButtonListener implements ActionListener
 {
 public void actionPerformed (ActionEvent event)
 {
 if(event.getSource() == greenButton)
 {
 message.setForeground(Color.green);
 }
 else if (event.getSource() == blueButton)
 {
 message.setForeground(Color.blue);
 }
 else if (event.getSource() == cyanButton)
 {
 message.setForeground(Color.cyan);
 }
 }
 }

 /\*\*Create a window and make it visible on the screen \*/
 public static void main(String[] args)
 {
 ColorFactory labGUI = new ColorFactory();
 labGUI.setVisible(true);
 }
}