

Programming Fundamentals II

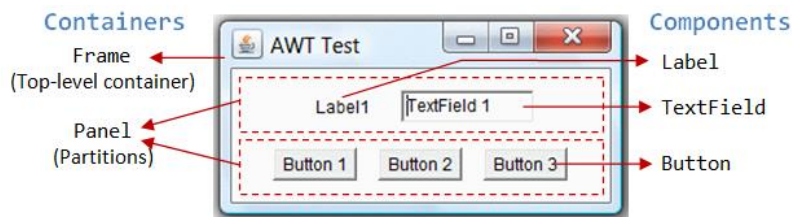
Lap8: - GUI
 - Event

Graphical User Interface: GUI

การเขียนโปรแกรมภาษา Java ให้มีการตอบสนองความต้องการของผู้ใช้งาน ทั้งทางด้านการรับข้อมูลจากผู้ใช้งาน ด้านความน่าสนใจ รวมทั้งความง่ายต่อการใช้งาน GUI ของโปรแกรมภาษา Java สามารถใช้ได้ทั้ง AWT และ Swing โดย Swing จะมีความแตกต่างระหว่าง AWT คือ Swing จะมีตัว J นำหน้าออบเจกต์ จาก Frame ดั้งเดิมของ AWT จะกลายเป็น JFrame เป็นออบเจกต์ของ Swing

ใน Swing จะมี Object frame ที่เรียกว่า JFrame ถูกเก็บในส่วน container บนสุด (Top level containers) ที่สามารถแสดงผลออกสู่หน้าจอได้ และส่วนที่เป็น panel (Non-top level containers) ที่ไม่สามารถแสดงผลได้ แต่จะมี JPanel, JScrollPane, JLayeredPane, JSplitPane และ JTabbedPane ทำหน้าที่ในการควบคุมการทำงานของเฟรม และ ส่วน simply เป็นส่วนด้านข้อความ เช่น text fields ต่างๆ

ส่วนติดต่อกับผู้ใช้แบบกราฟิก (GUI) ด้วยจาวา สามารถทำการแบ่งโครงสร้างหน้าเว็บที่แสดงข้อมูลและรับข้อมูลนั้น มีส่วนประกอบหลัก 2 ส่วน



1. Containers เป็นส่วนที่ทำหน้าที่ในเก็บ Components ต่างๆ ที่ต้องแสดงหน้าเว็บ ตามที่กำหนด
2. Components เป็นส่วนที่แสดงข้อมูล บนหน้าจอเว็บ เช่น Button, Label, and TextField เป็นต้น

ในการโปรแกรม GUI ทุก component ต้องถูกวางใน container โดยทุก component สามารถวางด้วยเมธอด add (Component c) มีผลทำให้ Container สามารถเข้ามาทำงานได้ แสดงตัวอย่างดังนี้

```
Panel panel = new Panel();           // Panel is a Container
Button btn  = new Button("Press");   // Button is a Component
panel.add(btn);                       // Panel Container adds a Button Comp.
```

JFrame

การสร้าง JFrame สามารถทำการเขียนด้วยการสร้าง frame มาหนึ่งเฟรมด้วยเมธอดที่ได้รับการสืบทอดมาจาก JFrame ซึ่งได้มาจากการ import javax.swing.* และสั่งปิดวินโดว์ได้ได้ด้วย Method setDefaultCloseOperation ดังแสดงตัวโปรแกรม

การออกแบบโปรแกรมสมัยใหม่ ควรออกแบบขนาด frame มาตรฐาน คล้ายกับขนาดของ Video ดังนี้

- 480 x 360
- 858 x 480
- 1280 x 720 (HD)
- 1920 x 1080 (Full HD)

Lab 8.1 ให้นักนิสิตสร้าง Class ชื่อ Lab81JFrame ทดลองสร้าง JFrame และสังเกตผลการทำงานของโปรแกรม Lab81JFrame.java เขียนโค้ดดังนี้

```
import javax.swing.*;

public class Lab81JFrame
{
    public static void main(String[] args)
    {
        JFrame frame = new JFrame("My Frame");
        frame.setSize(858, 480);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(frame.EXIT_ON_CLOSE);
    }
}
```

JPanel

การสร้าง JPanel นั้นเป็นการสร้าง Layout ของการจัด component แต่ละ Object ว่าต้องการให้มีส่วนประกอบใดบ้าง และส่วนประกอบใดทำงานก่อนหลังตามชนิดของ manager ที่กำหนดลงในรูปแบบคำสั่ง ถ้าไม่กำหนดค่าของ manager จะถูกกำหนดให้เป็น flow layout นั้นหมายความว่าในการจัดวาง layout จะทำการเรียงไปตามลำดับที่สั่งมา

Lab 8.2 ให้นักนิสิตสร้าง Class ชื่อ Lab82JPanel ทดลองสร้าง JFrame และ JPanel โดยให้กำหนดสีพื้นหลัง เป็นสีเทาเข้ม ให้นักนิสิตเขียนโค้ดจาก Lab82JPanel ลงด้านล่าง

```
import java.awt.Color;
import javax.swing.*;

public class Lab82JPanel
{
    public static void main(String[] args)
    {
        JFrame frame = new JFrame("My Frame");
        frame.setSize(300, 200);

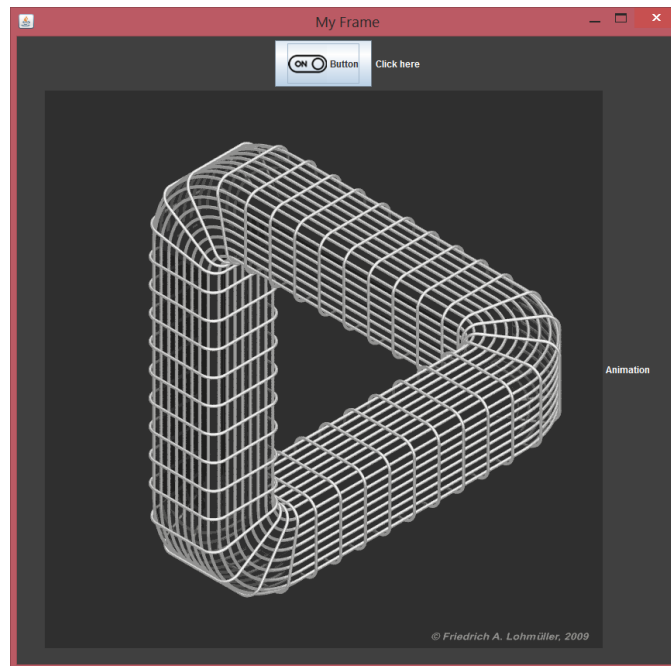
        JPanel panel = new JPanel(); // Create Panel
        panel.setBackground(Color.BLACK); // Set Color BG
        frame.add(panel); // Add Panel into Container

        frame.setVisible(true);
        frame.setDefaultCloseOperation(frame.EXIT_ON_CLOSE);
    }
}
```

Component (Button + Label)

Components เป็นส่วนประกอบเป็นส่วนที่แสดงข้อมูล บนหน้าจอ มาทดลองสร้าง component พื้นฐาน 2 ตัวคือ Button ปุ่มกด และ Label แถบอักษร

Lab 8.3 ให้นักนิสิตสร้าง Class ชื่อ Lab83Component และให้โหลดไฟล์ภาพตัวอย่างจาก folder (<https://goo.gl/3T5MzR>) และให้ทดลองเขียน component ดังตัวอย่างที่แสดง



ให้เขียนโค้ดจาก Lab83Component ลงด้านล่าง

```
import java.awt.*;
import javax.swing.*;

public class Lab83Component
{
    public static void main(String[] args)
    {
        JFrame frame = new JFrame("My Frame");
        frame.setSize(858, 858);

        JPanel panel = new JPanel(); // Create Panel
        panel.setBackground(Color.DARK_GRAY); // Set Color BG

        Icon icon1 = new ImageIcon("pic/button.png");
        Icon icon2 = new ImageIcon("pic/icon.gif");
        JButton btn1 = new JButton("Button", icon1);

        JLabel label1 = new JLabel("Click here", SwingConstants.RIGHT);
        JLabel label2 = new JLabel("Animation", icon2, SwingConstants.LEFT);

        label1.setForeground(Color.WHITE);
        label2.setForeground(Color.WHITE);

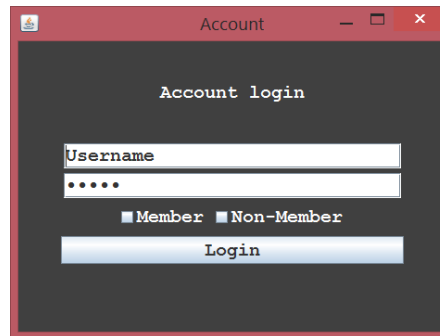
        panel.add(btn1); // add component1 btn
        panel.add(label1); // add component2 label1
        panel.add(label2); // add component3 label2

        frame.add(panel); // Add Panel into Container
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

Component (TextField + PasswordField + Cleckbox)

มาทดลองสร้าง component พื้นฐานเพิ่มเติมอีก 2 ตัวคือ textfield และ passwordfield

Lab 8.4 ให้นักศึกษาสร้าง Class ชื่อ Lab84Component และให้ทดลองเขียน component ดังตัวอย่างที่แสดง



ให้นักศึกษาเขียนโค้ดจาก Lab84Component ลงด้านล่าง

```
import java.awt.*;
import javax.swing.*;

public class Lab84Component {
    public static void main(String[] args) {
        Font font1 = new Font("Courier New", Font.BOLD, 20);
        JFrame frame = new JFrame("ACCOUNT");
        frame.setSize(480, 360);

        JPanel panel = new JPanel(); // Create Panel
        panel.setBackground(Color.DARK_GRAY); // Set Color BG

        JLabel label1 = new JLabel("Account login",
                                   SwingConstants.CENTER); // Create label1
        label1.setFont(font1);
        label1.setPreferredSize(new Dimension(480, 100));
        label1.setForeground(Color.white);

        JTextField text1 = new JTextField("Username", 30);
        text1.setFont(font1);

        JPasswordField pass1 = new JPasswordField("12345", 30);
        pass1.setFont(font1);

        JCheckBox checkbox1 = new JCheckBox("Member");
        checkbox1.setFont(font1);
        checkbox1.setBackground(Color.DARK_GRAY);
        checkbox1.setForeground(Color.WHITE);

        JCheckBox checkbox2 = new JCheckBox("Non-Member");
        checkbox2.setFont(font1);
        checkbox2.setBackground(Color.DARK_GRAY);
        checkbox2.setForeground(Color.WHITE);

        JButton btn1 = new JButton("Login");
        btn1.setFont(font1);
        btn1.setPreferredSize(new Dimension(370, 30));

        panel.add(label1);
        panel.add(text1);
        panel.add(pass1);
        panel.add(checkbox1);
        panel.add(checkbox2);
        panel.add(btn1);
        frame.add(panel);
        frame.setVisible(true);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    }
}
```

Layout Manager

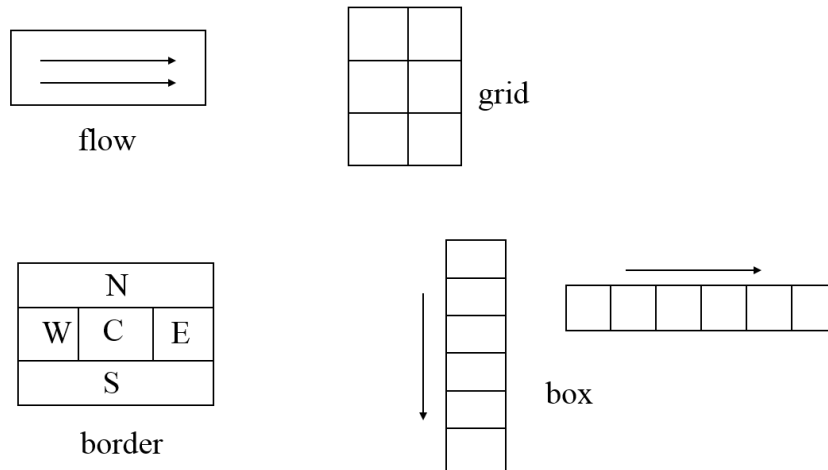
ในการจัดพื้นที่ส่วนของหน้าต่างที่ติดต่อกับผู้ใช้งานเป็นสิ่งที่ต้องให้มีการสื่อสารที่ตรงกันระหว่างตัวระบบกับผู้ใช้งาน ซึ่งการจัดองค์ประกอบคอมโพเนนต์ (Component) ลงบนพาเนล (panel)

FlowLayout เป็นพื้นฐานที่ได้ถูกกำหนดมาเป็นค่ามาตรฐานหรือ default นั้นเป็นการจัดวางองค์ประกอบให้ทำการแสดงที่เรียงลำดับตามบรรทัดแรกให้เต็มบรรทัดก่อนจึงทำการให้แสดงในบรรทัดถัดไป จนครบองค์ประกอบที่ถูกสั่งให้ทำการแสดง

BorderLayout เป็นการจัดวางองค์ประกอบให้ทำการแสดง ตามทิศทั้ง 5 ตำแหน่ง คือ ทิศเหนือ (North) ทิศใต้ (South) ทิศตะวันออก (East) ทิศตะวันตก (West) และจัดให้อยู่กึ่งกลาง

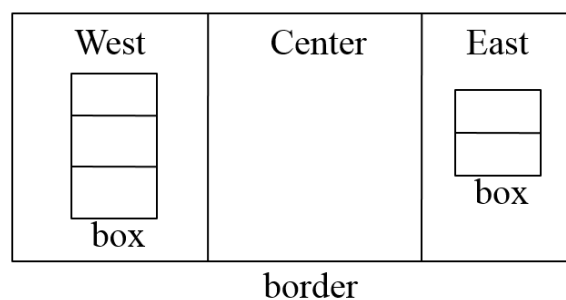
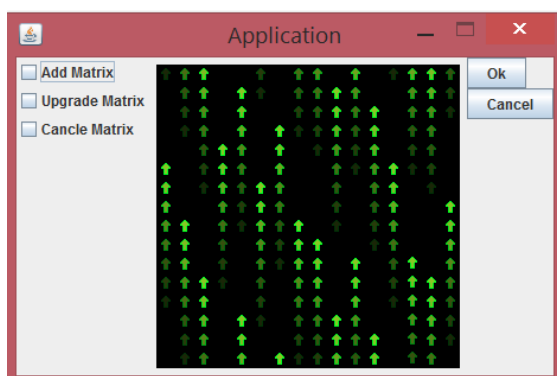
GridLayout เป็นการจัดวางองค์ประกอบให้ทำการแสดงในช่อง (grid) เป็นลักษณะตารางที่ประกอบด้วยแถวและคอลัมน์ โดยทำการจัดวางการแสดงผลจากด้านบนลงด้านล่าง และจากด้านซ้ายไปด้านขวา

BoxLayout เป็นการจัดวางในลักษณะเรียงกันต่อเนื่อง สามารถกำหนดการเรียงกันเป็น แนวตั้งและแนวนอน ลักษณะจะคล้ายๆกับ Flowlayout



ในส่วนของ Layout Manager ให้นิสิตต้องลองฝึกการออกแบบ layout ด้วยตนเอง เนื่องจากการออกแบบ layout เสมือนกับการออกแบบไม่มีขั้นตอนตายตัว

Lab 8.5 ให้นิสิตสร้าง Class ชื่อ Lab85Layout นิสิตจะได้ทดลองสร้าง Application ที่ใช้ Layout manger 2 Layout รวมกัน คือ Border + Box



ให้นักศึกษาเขียนโค้ดจาก Lab85Layout ลงด้านล่าง

```
import java.awt.*;
import javax.swing.*;

public class Lab85Layout extends JFrame
{
    public Lab85Layout()
    {
        super("_____ Application _____");

        Container c = getContentPane();

        // panel 1: vertical box layout
        JPanel p1 = new JPanel();
        p1.setLayout( new BoxLayout( p1, BoxLayout.Y_AXIS));
        JCheckBox check1 = new JCheckBox("Add Matrix");
        JCheckBox check2 = new JCheckBox("Upgrade Matrix");
        JCheckBox check3 = new JCheckBox("Cancle Matrix");

        p1.add(_____ check1 _____);

        p1.add(_____ check2 _____);

        p1.add(_____ check3 _____);

        // panel 2: vertical box layout
        JPanel p2 = new JPanel();
        p2.setLayout( new BoxLayout( p2, BoxLayout.Y_AXIS));
        JButton btn1= new JButton("Ok");
        JButton btn2 = new JButton("Cancel");

        p2.add(_____ btn1 _____);

        p2.add(_____ btn2 _____);

        // Label 1 : Center Application
        JLabel label1 = new JLabel( new
            ImageIcon("images/03_Animation.gif"));

        // add panels and image to frame
        c.add(p1, BorderLayout._____ WEST _____);

        c.add(label1, BorderLayout._____ CENTER _____);

        c.add(p2, BorderLayout._____ EAST _____);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        pack();
        setResizable(false);    // disable window resizing
        setLocationRelativeTo(null);
        setVisible(true);
    }

    public static void main(String[] args)
    {
        new _____ Lab85Layout() _____ ;
    }
}
```

Event Handling

จากการเขียนส่วนติดต่อกับผู้ใช้งานข้างต้น เป็นการสื่อสารระหว่างตัวโปรแกรมกับผู้ใช้งานเบื้องต้นที่ทำให้ผู้ใช้งานได้เห็นการแสดงผล ข้อมูลของตัวโปรแกรมที่ต้องการให้ผู้ใช้งานได้ทำการกำหนดข้อมูลให้กับตัวโปรแกรม นำไปทำการประมวลผลต่อ นั้น ดังนั้นการทำงานของโปรแกรมคอมพิวเตอร์ หลังจากได้ทำการสร้างหน้าต่างให้กับผู้ใช้งานแล้ว งานที่สำคัญอย่างยิ่งต่อจากนี้คือ การตรวจสอบการตอบสนองของผู้ใช้งานว่าได้ทำการสื่อสารกับมาได้การ คลิกเมาส์ หรือการกรอกข้อความและทำการกด Enter หรือยัง ดังนั้นตัวโปรแกรมต้องทำการสร้างออบเจกต์มาตรวจเหตุการณ์ (Event) อย่างต่อเนื่องตลอดเวลา ทำหน้าที่คอยเฝ้าระวังหรือคอยฟัง (Listener) ว่ามีการตอบสนองทางเมาส์หรือคีย์บอร์ดเมื่อใด จึงเรียกกระบวนการรวมๆ นี้ว่า Event Listener .

ตัวอย่าง ActionListener จากปุ่มกด JRadioButton ดังนี้

```

JRadioButton rB = new JRadioButton ("Single Choice");

rB.addActionListener ( new ActionListener ()
{
    public void actionPerformed ((ActionEvent event) )
    {

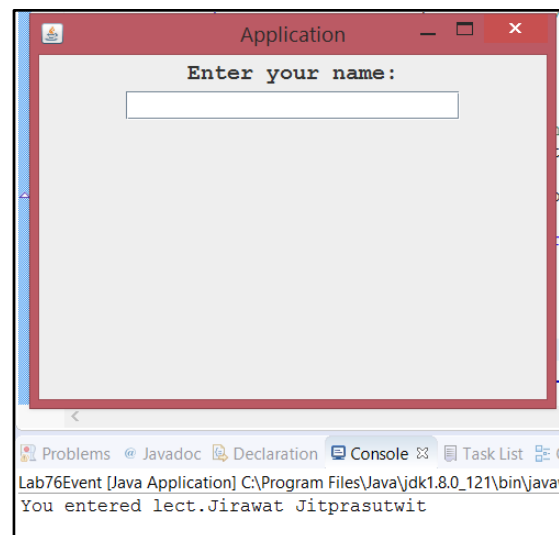
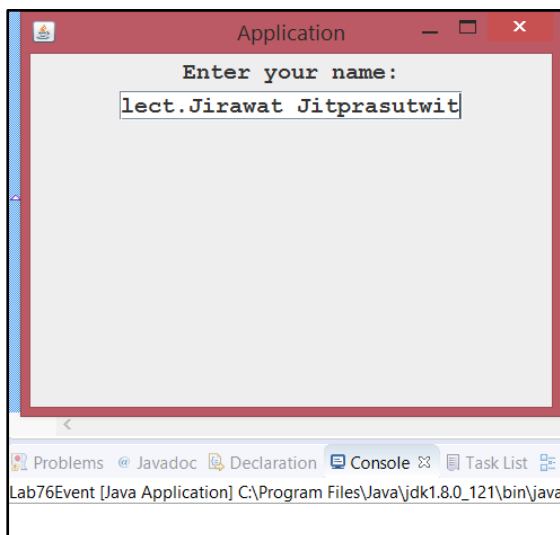
        JRadioButton rB = ( JRadioButton ) event.getSource ( );

        boolean isSelected = rB.isSelected ( );

        if ( isSelected )
            JOptionPane.showMessageDialog (null,"Click to selected." );
        else
            JOptionPane.showMessageDialog (null,"Click to unselected.");
    }
} );

```

Lab 8.6 ให้นิสิตสร้าง Class ชื่อ Lab86Event นิสิตจะได้ทดลองสร้าง Application ที่มี 1Label + 1Textfield เมื่อกรอกข้อความลงใน textfield แล้วกดปุ่ม enter ที่ keyboard ข้อความที่พิมพ์จะถูกพิมพ์ที่ Console และข้อความที่ Textfield จะถูก Clear ออก (ดู ตย จากรูปด้านล่าง)



ให้นิสิตเขียนโค้ดจาก Lab86Event ลงด้านล่าง

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class Lab86Event extends JFrame
{
    private JTextField textField1;    // global in actionPerformed()

    public Lab86Event()
    {
        super( "Application" );

        Container c = getContentPane();

        c.setLayout( new FlowLayout() );

        // label and text entry field
        Font font1 = new Font("Courier New", Font.BOLD, 20);
        JLabel label1 = new JLabel("Enter your name:");
        label1.setFont(font1);
        textField1 = new JTextField(25);    // 25 chars wide
        textField1.setFont(font1);

        c.add( label1 );

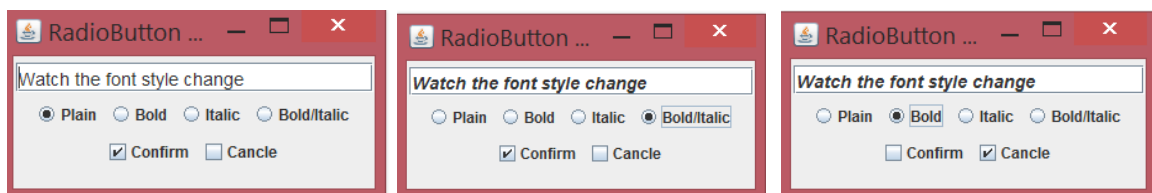
        c.add( textField1 );

        // Handle events from pressing return
        textField1.addActionListener( new ActionListener()
        {
            public void actionPerformed( ActionEvent event )
            {
                System.out.println("You entered " +
                    textField1.getText() );

                textField1.setText( "" );
            }
        } );
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(480,360);
        setVisible(true);
    }

    public static void main(String[] args)
    {
        new Lab86Event();
    }
}
```

Lab 8.7 ให้นิสิตสร้าง Class ชื่อ Lab87Event นิสิตจะได้ทดลองสร้าง Application ที่มี 1Textfield + 4RadioButton + 2Checkbox



Radio Button จะเลือกได้ 1 แบบเพื่อเลือกลักษณะ Fonts เพื่อไปแสดงลักษณะ Font ที่ TextField โดยจะเปลี่ยนลักษณะได้ต่อเมื่อ Checkbox อยู่ตำแหน่ง Confirm

ให้เขียนโค้ดจาก Lab87Event ลงด้านล่าง

```
import java.awt.*;
import javax.swing.*;
import java.awt.event.*;

public class Lab87Event extends JFrame
{
    private JTextField t;
    private Font plainFont, boldFont, italicFont, boldItalicFont;
    private JRadioButton plain, bold, italic, boldItalic;
    private JCheckBox check1, check2;
    private int checkBoxHandler = 1;

    public Lab87Event()
    {
        super( "RadioButton Test" );
        Container c = getContentPane();
        c.setLayout( new FlowLayout() );

        // 1 Text Field
        t = new JTextField("Watch the font style change", 25 );
        c.add( t );

        // 4 Radio Buttons
        plain = new JRadioButton( "Plain", true );

        bold = new JRadioButton( "Bold", false );

        italic = new JRadioButton( "Italic", false );

        boldItalic = new JRadioButton("Bold/Italic", false );

        c.add( plain ); c.add( bold );

        c.add( italic ); c.add( boldItalic );

        // 2 check boxes
        check1 = new JCheckBox("Confirm", true );
        check2 = new JCheckBox("Cancel");
        c.add( check1 ); c.add( check2 );

        // register all controls with 1 handler
        ButtonHandler handler = new ButtonHandler();
        plain.addItemListener( handler );
        bold.addItemListener( handler );
        italic.addItemListener( handler );
        boldItalic.addItemListener( handler );

        ButtonGroup radioGroup = new ButtonGroup();

        radioGroup.add( plain );      radioGroup.add( bold );

        radioGroup.add( italic );      radioGroup.add( boldItalic );

        ButtonGroup checkboxGroup = new ButtonGroup();

        checkboxGroup.add( check1 );

        checkboxGroup.add( check2 );

        plainFont = new Font( "TimesRoman", Font.PLAIN, 14 );
        boldFont = new Font( "TimesRoman", Font.BOLD, 14 );
        italicFont = new Font( "TimesRoman", Font.ITALIC, 14 );
        boldItalicFont = new Font( "TimesRoman", Font.BOLD + Font.ITALIC, 14 );

        t.setFont( plainFont );
    }
}
```

```

        // itemListener for check box1 Confirm
        check1.addItemListener( new ItemListener() {
            public void itemStateChanged(ItemEvent e){
                if (e.getStateChange() == e.SELECTED){
                    System.out.print("selected ");
                    checkBoxHandler = 1;
                }
                else
                    System.out.print("de-selected ");
                System.out.print("Confirm\n");
            }
        }
    );

    // itemListener for check box2 Cacle
    check2.addItemListener( new ItemListener(){
        public void itemStateChanged(ItemEvent e){
            if (e.getStateChange() == e.SELECTED){
                System.out.print("selected ");
                checkBoxHandler = 0;
            }
            else
                System.out.print("de-selected ");
            System.out.print("Cacle\n");
        }
    }
    );

    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setSize(300,150);
    setVisible(true);
}

public static void main(String[] args)
{
    new Lab87Event();
}

class ButtonHandler implements ItemListener
{
    public void itemStateChanged( ItemEvent e )
    {
        if (e.getStateChange() == ItemEvent.SELECTED)
        {
            if( checkBoxHandler == 1 )
            {
                if ( e.getSource() == plain )
                    t.setFont( plainFont );

                else if ( e.getSource() == bold )
                    t.setFont( boldFont );

                else if ( e.getSource() == italic )
                    t.setFont( italicFont );

                else if ( e.getSource() == boldItalic )
                    t.setFont( boldItalicFont );

            }
            t.repaint();
        }
    }
}
}

```

Homework Lab8

GUI#1

1. ช่องและปุ่ม

จงเขียนโปรแกรมให้มี

1. text field หนึ่งช่อง
2. button หนึ่งปุ่ม เขียนว่า “Clear”

เมื่อกรอกข้อความใน text field แล้วกดปุ่ม Enter ที่ Keyboard โปรแกรมจะแสดงข้อความนั้นทาง message dialog แต่ถ้าคลิกที่ button โปรแกรมจะลบข้อความใน text field (แนะนำให้เขียน action listener สองตัวแยกกัน สำหรับ text field ตัวหนึ่ง และสำหรับ button อีกตัวหนึ่ง)

2. สลับไปสลับมา

จงเขียนโปรแกรมซึ่งมีองค์ประกอบดังนี้

1. มี text field สองช่อง
2. มี check box 2 รายการ รายการแรกเขียนว่า Swap รายการที่สองเขียนว่า Modify on Swap
3. มี label หนึ่งอัน
4. มี button สองปุ่ม ปุ่มแรกเขียนว่า Submit ปุ่มที่สองเขียนว่า Clear

เมื่อคลิกปุ่ม Submit โปรแกรมจะนำข้อความจากทั้งสอง text field มาต่อกันแล้วแสดงที่ label แต่ถ้าคลิกปุ่ม Clear โปรแกรมจะลบข้อความจากทั้งสอง text field และที่ label

ส่วน check box ทั้งสองรายการมีผลดังนี้ check box Swap ถ้าเลือกไว้ เมื่อคลิกปุ่ม Submit โปรแกรมจะนำข้อความมาต่อกันแล้วแสดงที่ label แต่สลับตำแหน่งเอาข้อความแรกไปต่อท้ายข้อความหลัง ถ้า check box Modify on Swap ถูกเลือกด้วย จะมีการสลับข้อความใน text field ทั้งสองด้วยจริง ๆ แต่ถ้าเลือก Modify on Swap แต่ไม่ได้เลือก Swap ไว้ ให้ถือว่าไม่มีผลใด ๆ (เหมือนไม่ได้เลือกทั้งคู่)

GUI#1

```

/*
 * Written by Nititorn Kijprasopchok
 * ID: 6530300295
 */

import java.awt.*;
import javax.swing.JTextField;
import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

public class ClearButton extends JFrame{
    private JTextField textField;
    private JButton clearBtn;
    public ClearButton(){
        super("Clear Button");
        Container c = getContentPane();
        c.setLayout(new FlowLayout());

        //create textfield
        textField = new JTextField(20);
        c.add(textField);

        //create button
        clearBtn = new JButton("Clear");
        c.add(clearBtn);

        //event handler for textfield
        textField.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent event){
                System.out.println(textField.getText());
            }
        });

        //event handler for button
        clearBtn.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent event){
                textField.setText("");
            }
        });

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(300, 100);
        setLocationRelativeTo(null);
        setVisible(true);
    }

    public static void main(String[] args){
        new ClearButton();
    }
}
```

GUI#2

```
/*
 * Written by Nititorn Kijprasopchok
 * ID: 6530300295
 */

import javax.swing.*;
import java.awt.*;
import java.awt.event.*;

public class SwapButton extends JFrame{
    private JTextField textField1, textField2, textFieldTemp;
    private JCheckBox check1, check2;
    private JLabel label;
    private JButton btn1, btn2;
    private int checkBoxHandler1 = 0, checkBoxHandler2 = 0;

    public SwapButton(){
        super("Swap Button");
        Container c = getContentPane();
        c.setLayout(new FlowLayout());

        //2 text field
        textField1 = new JTextField(25);
        textField2 = new JTextField(25);
        c.add(textField1); c.add(textField2);

        //1 label
        label = new JLabel("[LABEL HERE]", SwingConstants.CENTER);
        label.setPreferredSize(new Dimension(200, 50));
        c.add(label);

        //2 check box
        check1 = new JCheckBox("Swap");
        check2 = new JCheckBox("Modify on Swap");
        c.add(check1); c.add(check2);

        //2 button
        btn1 = new JButton("Submit");
        btn2 = new JButton("Clear");
        c.add(btn1); c.add(btn2);

        //event handler for check box 1
        check1.addItemListener(new ItemListener(){
            public void itemStateChanged(ItemEvent event){
                if(event.getStateChange() == event.SELECTED){
                    checkBoxHandler1 = 1;
                }else{
                    checkBoxHandler1 = 0;
                }
            }
        });

        //event handler for check box 2
        check2.addItemListener(new ItemListener(){
            public void itemStateChanged(ItemEvent event){
                if(event.getStateChange() == event.SELECTED){
                    checkBoxHandler2 = 1;
                }else{
                    checkBoxHandler2 = 0;
                }
            }
        });

        //event handler for submit button
        btn1.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent event){
                if(checkBoxHandler1 == 1 && checkBoxHandler2 == 0){
                    textFieldTemp = textField1;
                    textField1 = textField2;
                    textField2 = textFieldTemp;
                    label.setText(textField1.getText() + textField2.getText());
                }else{
                    label.setText(textField1.getText() + textField2.getText());
                }
            }
        });

        //event handler for clear button
        btn2.addActionListener(new ActionListener(){
            public void actionPerformed(ActionEvent event){
                textField1.setText("");
                textField2.setText("");
                label.setText("");
            }
        });

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setSize(250, 250);
        setLocationRelativeTo(null);
        setVisible(true);
    }

    public static void main(String[] args){
        new SwapButton();
    }
}
```