LAB 8 – Java GUI

Objectives:

At the end of this lab, the students are able to:

i. Use *JOptionPane* for creating a simple GUI input/output.
ii. Using basic Java Swing's components to develop simple GUI.
iii. Demonstrate the use of nested class to implement even handler for triggering the action on
iv. Swing's components

8.1 Activity 1

8.1.1 Objective

Demonstrate how to use basic Java Swing's components to develop simple GUI.

8.1.2 Problem Description

Create the following GUI (as shown in Figure 1) using related Java Swing's component. You do not have to provide any functionality.

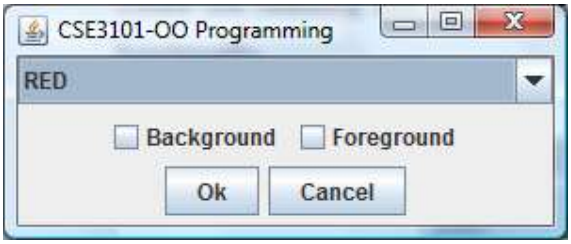

Figure 1

[Estimated Time: 40 minutes]

8.1.3 Solution Activity 1

Algorithms
Step 1 – Declare Java Swing and Awt API. Step 2 – Inherit JFrame class.
Step 3 – Define reference variables for button, checkbox, combo box and panel
Step 4 – Implement logic inside the constructor.
4.1 – Label the frame
4.2 – Define the layout
4.3 – Instantiate class JComboBox() and add an item as RED.
4.4 – Attach JComboBox to layout.
4.5 – Instantiate class JPanel()
4.6 – Instantiate class JCheckBox() and name as Background and Foreground
4.7 – Attach item JCheckBox to the panel.
4.8 – Attach panel to the layout.

4.9 – Instantiate class JButton() to represent Cancel and OK button
4.10 - Instantiate class JPanel()
4.11 – Attach item JButton to the panel.
4.12 – Attach panel to the layout. Step 5 – Instantiate class ColorSelect
Step 5 – Instantiate class ColorSelect.

Programming

Step 1: Go to CSF3043 folder and create sub-folder called as Lab – Chapter 7.

Step 2: Go to CSF3043 –> Lab – Chapter 7's folder and create Sub-working folder called Activity 1.

Step 3: Open IDE NetBeans or JCreator.

Step 4: Create new file/class called ColorSelectFrame.java

Step 5: Import Java Swing and AWT component.

```
/**
 * ------------------------------------------------------------------------
  * @author        : Mohamad Nor Hassan
  * Program Name   : ColorSelectFrame
  * Description    : To creates a simple GUI using Java Swing's component.
  * Creation Date  : 06 Sept 2011
  * Modified Date  : None
  * Version        : Version 1.00
  * ------------------------------------------------------------------------
  */
import java.awt.BorderLayout;
import javax.swing.JFrame;
import javax.swing.JButton;
import javax.swing.JCheckBox;
import javax.swing.JComboBox;
import javax.swing.JPanel;
```

Step 6: Define the class *ColorSelectFrame()* and instance variables as below:

```
public class ColorSelectFrame extends JFrame
{
   //Define all reference variables that represent components of GUI
   private JButton okJButton;
   private JButton cancelJButton;
   private JCheckBox backgroundJCheckBox;
   private JCheckBox foregroundJCheckBox;
   private JComboBox colorJComboBox;
   private JPanel panel;
   private JPanel panel2;
```

Step 7: Define and implement the logic for *ColorSelectFrame()* constructor as below:

```
   // Implement logic inside constructor - To creates components and add them to applet
   public ColorSelectFrame()
   {
      super( "CSE3101-OO Programming" );
      setLayout( new BorderLayout() );

      colorJComboBox = new JComboBox(); // create combobox
      colorJComboBox.addItem( "RED" ); // add red item
      add( colorJComboBox, BorderLayout.NORTH ); // put in north region

      panel = new JPanel(); // create a panel
      backgroundJCheckBox = new JCheckBox( "Background" ); // background
      foregroundJCheckBox = new JCheckBox( "Foreground" ); // forground
      panel.add( backgroundJCheckBox ); // add background checkbox
      panel.add( foregroundJCheckBox ); // add foreground chechbox
      add( panel, BorderLayout.CENTER ); // add panel to center region

      okJButton = new JButton( "Ok" ); // create ok button
      cancelJButton = new JButton( "Cancel" ); // create cancel button
      panel2 = new JPanel(); // create panel to hold buttons
      panel2.add( okJButton ); // add ok button to panel
      panel2.add( cancelJButton ); // add cancel button to panel
      add( panel2, BorderLayout.SOUTH ); // add panel to south region
   } // end ColorSelectFrame constructor
} // end class ColorSelectFrame
```

Step 8: Compile the ColorSelectFrame.java.

Step 9: Open new file/class called ColorSelect.java

Step 10: Instantiate *ColorSelectFrame()* class.

```
/* ------------------------------------------------------------------------
 * Author         : Mohamad Nor Hassan
 * Program Name   : ColorSelect
 * Description    : To execute class ColorSelect()
 * Creation Date  : 06 Sept 2011
 * Modified Date  : None
 * Version        : Version 1.00
 * ------------------------------------------------------------------------
 */
import javax.swing.JFrame;

public class ColorSelect
{
   public static void main ( String args[] )
   {
      ColorSelectFrame colorSelectFrame = new ColorSelectFrame();
      colorSelectFrame.setDefaultCloseOperation( JFrame.EXIT_ON_CLOSE );
      colorSelectFrame.setSize( 300, 125 );
      colorSelectFrame.setVisible( true );
   }
}
```

Step 11: Compile your program and evaluate the output.

8.2 Activity 2

8.3.1 Objective

To implement simple GUI input/output using *JOptionPane.*

8.3.2 Problem Description

Write a program by implementing GUI components using class *JOptionPane* . Your program should be able to do the following tasks:

- Accept the input from end user for width and length of rectangle.
- Calculate the area and perimeter of the rectangle.
- Display the area and perimeter for rectangle.

Note: Use OOP approach for your programming

[Estimated Time: 40 minutes]

8.3 Activity 3

8.3.1 Objective

Demonstrate the use of nested classes to implement event handler for triggering the action on Swing components.

8.3.2 Problem Description

Write a temperature conversion application that convert from Fahrenheit to Celcius. The Fahrenheit temperature should be entered from the keyboard (via a JTextField). A JLabel should be used to display the converted temperature. Use the following formula for the conversion:

$$Celsius = 5/9 \times (Fahrenheit - 32)$$

The sample of the GUI is shown in Figure 2.

Figure 2

[Estimated Time: 40 minutes]

8.3.3 Solution Activity 3

Algorithms
Step 1 – Declare Java Swing and Awt API.
Step 2 – Inherit JFrame class.
Step 3 – Define reference variables for label and textfield
Step 4 – Implement logic inside the constructor.
- 4.1 – Label the frame
- 4.2 – Instantiate class JTextField() and JLabel ()
- 4.3 – Define action listener as a nested class ActionListener()
- 4.4 – Create method ActionPerformed() and implement the logic.
- 4.5 – Set the layout
- 4.6 – Attach each component to the layout. Step 5 – Instantiate class ConvertFrame.

Step 5 – Instantiate class ConvertFrame.

Programming

Step 1: Go to CSF3043 –> Lab – Chapter 7 folder and create Sub-working folder called Activity 3.

Step 2: Open IDE Netbeans or JCreator.

Step 3: Create new file/class called ConvertFrame.java.

Step 4: Import Java Swing and AWT component.

```
/**
 * @(#)ConvertFrame.java
 * ---------------------------------------------------------------------
 * @author         : Mohamad Nor Hassan
 * @Program Name   : ConvertFrame
 * @Description    : GUI for temperature conversion.
 * @Creation Date  : 28 Feb 2011
 * @Modified Date  : None
 * @Version        : Version 1.00
 * ---------------------------------------------------------------------
 */
import java.awt.BorderLayout;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.JTextField;
```

Step 5: Define the class *ConvertFrame()* and instance variables as below:

```
public class ConvertFrame extends JFrame
{
   private JLabel label1; // label to prompt user to enter fahrenheit
   private JLabel label2; // label to display temperature in celsius
   private JTextField temperatureF; // textfield to enter temperature
```

Step 6: Define and implement the logic for *ConverFrame()* constructor as below:

```
    // Define constructer sets up GUI
    public ConvertFrame()
    {
       super( "Temperature converter" );

       label1 = new JLabel( "Enter Fahrenheit temperature:" );
       temperatureF = new JTextField( 10 ); // textfield for fahrenheit

       // register anonymous action listener
       temperatureF.addActionListener
       (
          new ActionListener() // anonymous inner class
          {
             public void actionPerformed(ActionEvent e)
             {
                int temp = Integer.parseInt( temperatureF.getText() );
                int celcius = ( int ) ( 5.0f / 9.0f * ( temp - 32 ) );
                label2.setText( "Temperature in Celcius is: " + celcius );
             } // end method actionPerformed
          } // end anonymous inner class
       ); // end call to addActionListener

       label2 = new JLabel( "Temperature in Celcius is:" );

       setLayout( new BorderLayout() );           // set use border layout
       add( label1, BorderLayout.NORTH );         // set north region
       add( temperatureF, BorderLayout.CENTER ); // set center region
       add( label2, BorderLayout.SOUTH );         // set south region
    }
}
```

Step 7: Compile the ConvertFrame.java.

Step 8: Open new file/class called Convert.java

Step 9: Instantiate ConvertFrame() class

Step 10: Compile your program and evaluate the output.

Lab Exercises

Objective

To create simple GUI and use an event handler to perform specific action

Problem Description

Create a simple GUI to change the type of font when end user point of specific radio button.

[Estimated Time: 60 minutes]