Advanced Java Programming Lab Sheet
III Year /VI Part
Faculty: BCA

Lab sheet 2

Objectives:

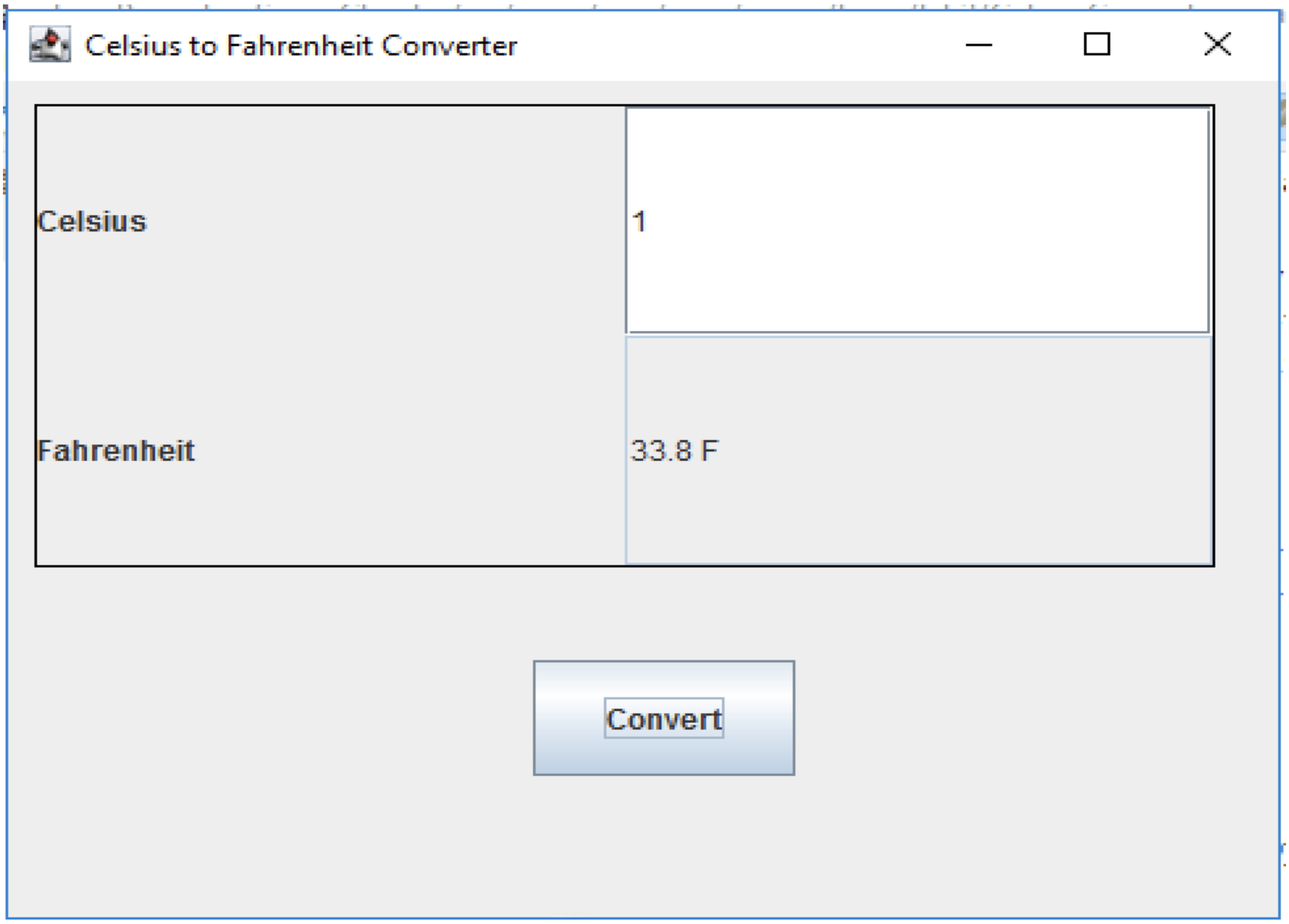
1. To Familiarized different types of event handling in swing such as Action Event, Mouse Event, Key Event and Window Event.
2. Displaying and Scaling Image with Swing GUI.
3. To understand the concept of Swing MVC Design.

Objective 1:

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JLabel;

public class ActionEventHandling extends JFrame implements ActionListener {

        private static final long serialVersionUID = 1L;
        int count = 0;
        JButton btn_click;
        JLabel lbl_result;

        public ActionEventHandling() {

                setTitle("Action Event Handling");
                setLayout(null);
                setBounds(10, 10, 500, 400);

                btn_click = new JButton("Click Me");
                btn_click.setActionCommand("Click");
                btn_click.setBounds(20, 20, 100, 50);
                btn_click.addActionListener(this);

                lbl_result = new JLabel("Result : 0");
                lbl_result.setBounds(20, 100, 100, 10);

                add(lbl_result);
                add(btn_click);

                setVisible(true);
                setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        }

        public void actionPerformed(ActionEvent e) {
                count = count + 1;
                String result = e.getActionCommand();

                if (result.equals("Click")) {
                        lbl_result.setText("Result : " + count);
                }
        }

        public static void main(String[] args) {

                new ActionEventHandling();
        }
}
```

Assignment:

1.1. Write a Java code and display the window with a JLabel, JButton and JTexField as shown in figure below. When a user clicks on a button, it should convert the value of Celsius to Fahrenheit and display the result in text Field.



Celsius 1

Fahrenheit 33.8 F

1.2. Write a Java code and display the window with registering mouse listeners. Handle the following mouse event:

1. When the user pressed the mouse, change the background color of the window to Red Color.
2. When the mouse enters the window, change the background color of the window to Green Color.
3. When the mouse exited from the window, change the background color of the window to Yellow Color.
4. When the user clicks on the mouse, change the background color of the window to Gray color.

Hint:

```
import java.awt.event.MouseEvent;
import java.awt.event.MouseListener;

import javax.swing.JFrame;

public class MouseEventHandling extends JFrame implements MouseListener {

	private static final long serialVersionUID = 1L;

	public MouseEventHandling() {}

	public void mouseClicked(MouseEvent e) {
		Container container =  this.getContentPane();
		container.setBackground(Color.GRAY);
	}

	public void mousePressed(MouseEvent e) {}

	public void mouseReleased(MouseEvent e) {}

	public void mouseEntered(MouseEvent e) {}

	public void mouseExited(MouseEvent e) {}

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

    }
```

1.3 Write a Java code and display the window with textField and register the text field with a KeyListner. Handle the following key events.

a. When user type b in the text field changes the background color of the window.
b. When a user presses any key, display the name of the key in the window.

1.4. Write a Java program and demonstrate the window event by registering the window as a window listener.

Objective 2:

```
import java.awt.Color;
import java.awt.Image;

import javax.swing.BorderFactory;
import javax.swing.ImageIcon;
import javax.swing.JFrame;
import javax.swing.JLabel;
import javax.swing.border.Border;

public class DisplayAndScaling extends JFrame {

	private static final long serialVersionUID = 1L;
	JLabel lblShowImage;

	public DisplayAndScaling() {

		setTitle("Displaying and Scaling Image");
		setLayout(null);
		setBounds(10, 10, 500, 400);

		Border bd = BorderFactory.createLineBorder(Color.red);

		lblShowImage = new JLabel();
		lblShowImage.setBounds(10, 10, 450, 200);
		lblShowImage.setBorder(bd);

		String image_path = "dog.jpg";
		ImageIcon imageIcon = new ImageIcon(image_path);
		Image image = imageIcon.getImage();

		ImageIcon resized_image = new ImageIcon(
				image.getScaledInstance(lblShowImage.getWidth(),
lblShowImage.getHeight(), Image.SCALE_SMOOTH));
		lblShowImage.setIcon(resized_image);

		add(lblShowImage);

		setVisible(true);
		setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

	}

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

    }
```

Assignment:

2.0. Write a Java program and create a window with 3 buttons having an icon on it. When a user clicks on the button, it should display the list of corresponding images.

Objective 3:

1. Create Controller:

```
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;


public class Controller {

    private Model model;
    private View view;
    private ActionListener actionListener;

    public Controller(Model model, View view){
        this.model = model;
        this.view = view;

    }

    public void contol(){
        actionListener = new ActionListener() {
              public void actionPerformed(ActionEvent actionEvent) {
                  linkBtnAndLabel();
              }
        };
        view.getButton().addActionListener(actionListener);
    }

    private void linkBtnAndLabel(){
        model.incX();
        view.setText(Integer.toString(model.getX()));
    }
   }
```

2. Create Model

```
public class Model {

     private int x;

     public Model() {
          x = 0;
     }
```

```
	public Model(int x) {
		this.x = x;
	}

	public void incX() {
		x++;
	}

	public int getX() {
		return x;
	}
}
```

3. Create view

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

public class View {

	private JFrame frame;
	private JLabel label;
	private JButton button;

	public View(String text) {
		frame = new JFrame("View");
		frame.getContentPane().setLayout(new BorderLayout());
		frame.setSize(400, 200);

		label = new JLabel(text);
		frame.getContentPane().add(label, BorderLayout.CENTER);

		button = new JButton("Button");
		frame.getContentPane().add(button, BorderLayout.SOUTH);

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

	public JButton getButton() {
		return button;
	}

	public void setText(String text) {
		label.setText(text);
	}
}
```

4. Create Main class to execute all Model, View and controller.

```
public class Main {

     public static void main(String[] args) {
          Model model = new Model(0);
          View view = new View("0");
          Controller controller = new Controller(model, view);
          controller.contol();
     }

   }
```

Assignment:

3.0. Explain about the swing MVC design, execute the above program, and list out the differences of swing mvc design and normal design (without swing mvc).