5COSC019W - Solutions to Tutorial 8 Exercises

1 A Simple Swing Program

This was described in detail in the lecture and the lecture notes.

2 Event Handling

1. This was described in detail in the lecture and the lecture notes.

2.
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;


// window event Handler class
class MyWindowListener extends WindowAdapter {
    public void windowClosing(WindowEvent e) {
        System.out.println("Closing window!");
        System.exit(0);
    }
}

// button event handler class
class MyActionListener implements ActionListener {
    private int i=1;
    JFrame frame;
    JButton b1;


    MyActionListener(JFrame f) {
        frame = f;
    }

    public void actionPerformed(ActionEvent e) {
        System.out.println("Pressed Button " + i++ + "th time!");

        if (i % 2 == 0)
            frame.getContentPane().setBackground(Color.red);
        else
```

```
            frame.getContentPane().setBackground(Color.white);
    }
}


class MyActionListener2 implements ActionListener {
    JFrame frame;

    MyActionListener2(JFrame f) {
        frame = f;

    }

    public void actionPerformed(ActionEvent e) {
        frame.getContentPane().setBackground(Color.yellow);
    }
}

public class ComponentExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("ComponentExample");
        JButton button = new JButton("press me");
        JButton button2 = new JButton("button 2");
        JPanel jp = new JPanel();
        jp.setBackground(Color.white);

        // set the content pane to be the newly created JPanel
        frame.setContentPane(jp);

        frame.getContentPane().add(button);
        frame.getContentPane().add(button2);

        // register an event handler for frame events
        frame.addWindowListener(new MyWindowListener());

        // register an event handler for button events
        MyActionListener listener = new MyActionListener(frame);
        button.addActionListener(listener);
        button2.addActionListener(new MyActionListener2(frame));

        frame.setSize(400, 400);
        frame.setVisible(true);
    }
}
```

3.
```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
```

```java
// window event Handler class
class MyWindowListener extends WindowAdapter {
    public void windowClosing(WindowEvent e) {
        System.out.println("Closing window!");
        System.exit(0);
    }
}

// button event handler class
class MyActionListener implements ActionListener {
    private int i=1;
    JFrame frame;
    JButton b1;

    MyActionListener(JFrame f, JButton b) {
        frame = f;
        b1 = b;
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == b1) {   // button 1 was the source of the event
            System.out.println("Pressed Button " + i++ + "th time!");

            if (i % 2 == 0)
                frame.getContentPane().setBackground(Color.red);
            else
                frame.getContentPane().setBackground(Color.white);
        }
        else // button 2 was the source of the event
            frame.getContentPane().setBackground(Color.yellow);
    }
}


public class ComponentExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("ComponentExample");
        JButton button = new JButton("press me");
        JButton button2 = new JButton("button 2");
        JPanel jp = new JPanel();
        jp.setBackground(Color.white);

        // set the content pane to be the newly created JPanel
        frame.setContentPane(jp);

        frame.getContentPane().add(button);
        frame.getContentPane().add(button2);
```

```
            // register an event handler for frame events
            frame.addWindowListener(new MyWindowListener());

            // register an event handler for button events
            MyActionListener listener = new MyActionListener(frame, button);
            button.addActionListener(listener);
            button2.addActionListener(listener);

            frame.setSize(400, 400);
            frame.setVisible(true);
        }
    }
```

3 Layout Managers

1.

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

// window event Handler class
class MyWindowListener extends WindowAdapter {
    public void windowClosing(WindowEvent e) {
        System.out.println("Closing window!");
        System.exit(0);
    }
}

public class LayoutManagersExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("SimpleSwingExample");
        JPanel panel = new JPanel();
        //panel.setLayout(new BorderLayout());
        panel.setLayout(new BoxLayout(panel, BoxLayout.X_AXIS));

        JButton b1 = new JButton("Button 1");
        JButton b2 = new JButton("Button 2");
        JButton b3 = new JButton("Button 3");
        JButton b4 = new JButton("Button 4");

        panel.add(b1);
        panel.add(b2);
        panel.add(b3);
        panel.add(b4);

        frame.setContentPane(panel);

        // register an event handler for frame events
        //frame.addWindowListener(new MyWindowListener());
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        frame.pack();
        frame.setVisible(true);
    }
}
```

2.
```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

// window event Handler class
class MyWindowListener extends WindowAdapter {
    public void windowClosing(WindowEvent e) {
        System.out.println("Closing window!");
        System.exit(0);
    }
}

public class LayoutManagersExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("SimpleSwingExample");
        JPanel panel = new JPanel();
         //panel.setLayout(new BorderLayout());
        panel.setLayout(new BoxLayout(panel, BoxLayout.X_AXIS));

        JButton b1 = new JButton("Button 1");
        JButton b2 = new JButton("Button 2");
        JButton b3 = new JButton("Button 3");
        JButton b4 = new JButton("Button 4");

        panel.add(b1);
        panel.add(b2);
        panel.add(b3);
        panel.add(b4);

        frame.setContentPane(panel);

        // register an event handler for frame events
        //frame.addWindowListener(new MyWindowListener());
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //frame.setSize(400, 400);
        frame.pack();
        frame.setVisible(true);
    }
}
```

3.
```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

// window event Handler class
class MyWindowListener extends WindowAdapter {
    public void windowClosing(WindowEvent e) {
        System.out.println("Closing window!");
        System.exit(0);
    }
}

public class LayoutManagersExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("SimpleSwingExample");
        JPanel panel = new JPanel();
         //panel.setLayout(new BorderLayout());
        panel.setLayout(new BoxLayout(panel, BoxLayout.Y_AXIS));

        JButton b1 = new JButton("Button 1");
        JButton b2 = new JButton("Button 2");
        JButton b3 = new JButton("Button 3");
        JButton b4 = new JButton("Button 4");

        panel.add(b1);
        panel.add(b2);
        panel.add(b3);
        panel.add(b4);

        frame.setContentPane(panel);

        // register an event handler for frame events
        //frame.addWindowListener(new MyWindowListener());
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //frame.setSize(400, 400);
        frame.pack();
        frame.setVisible(true);
    }
}
```

4. FlowLayout is the default layout for `JPanel`s but let's do the setting explicitly:

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

// window event Handler class
class MyWindowListener extends WindowAdapter {
    public void windowClosing(WindowEvent e) {
        System.out.println("Closing window!");
        System.exit(0);
    }
}

public class LayoutManagersExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("SimpleSwingExample");
        JPanel panel = new JPanel();
         //panel.setLayout(new BorderLayout());
        panel.setLayout(new FlowLayout());

        JButton b1 = new JButton("Button 1");
        JButton b2 = new JButton("Button 2");
        JButton b3 = new JButton("Button 3");
        JButton b4 = new JButton("Button 4");

        panel.add(b1);
        panel.add(b2);
        panel.add(b3);
        panel.add(b4);

        frame.setContentPane(panel);

        // register an event handler for frame events
        //frame.addWindowListener(new MyWindowListener());
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //frame.setSize(400, 400);
        frame.pack();
        frame.setVisible(true);
    }
}
```

5.
```
import javax.swing.*;
import java.awt.event.*;
import java.awt.*;

// window event Handler class
class MyWindowListener extends WindowAdapter {
    public void windowClosing(WindowEvent e) {
        System.out.println("Closing window!");
        System.exit(0);
    }
}

public class LayoutManagersExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("SimpleSwingExample");
        JPanel panel = new JPanel();
         //panel.setLayout(new BorderLayout());
        panel.setLayout(new GridLayout(2,2));

        JButton b1 = new JButton("Button 1");
        JButton b2 = new JButton("Button 2");
        JButton b3 = new JButton("Button 3");
        JButton b4 = new JButton("Button 4");

        panel.add(b1);
        panel.add(b2);
        panel.add(b3);
        panel.add(b4);

        frame.setContentPane(panel);

        // register an event handler for frame events
        //frame.addWindowListener(new MyWindowListener());
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        //frame.setSize(400, 400);
        frame.pack();
        frame.setVisible(true);
    }
}
```

4 JLabel and JTextField

1. This example was explained in the lecture.

2.
```
public void actionPerformed(ActionEvent e) {
        System.out.println("You entered: " + e.getActionCommand());
    }
}

class MyActionListener2 implements ActionListener {
    public void actionPerformed(ActionEvent e) {
        System.out.println("Second field contains: " + e.getActionCommand());
    }
}

public class LabelFieldExample2 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("JLabel and JTextField Example");

        JLabel label = new JLabel("Enter your name: ");
        // create a field with 25 chars width
        JTextField field = new JTextField(25);
        JTextField field2 = new JTextField(35);
```

```
            // put components next to each other in the x-direction
            Container c = frame.getContentPane();
            c.setLayout(new BoxLayout(c, BoxLayout.X_AXIS));

            // add label and field in the frame
            c.add(label);
            c.add(field);
            c.add(field2);

            // register an event handler for frame events
            frame.addWindowListener(new MyWindowListener());

            // register an event handler for button events
            field.addActionListener(new MyActionListener());
            field2.addActionListener(new MyActionListener2());

            //frame.setSize(400, 400);
            frame.pack();
            frame.setVisible(true);
        }
    }
```

5 Creating Professionally Looking Layouts

This was described in detail in the lecture.

6 Choosing a Colour

In this sample solution we create two separate frames to make the results more visible as the JColorChooser component occupies quite some space itself (by default) in a frame.

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


class MyActionListener implements ActionListener {
    JFrame frame;
    JColorChooser chooser;

    MyActionListener(JFrame f, JColorChooser chooser) {
        frame = f;
        this.chooser = chooser;
    }

    public void actionPerformed(ActionEvent e) {
        // get the chosen colour set by the user
        Color c = chooser.getColor();

        // set the background
        frame.getContentPane().setBackground(c);
    }
}

public class ColourChooser {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Colour Chooser");

        JButton button = new JButton("Change colour");


        JColorChooser chooser = new JColorChooser();

        // Create a separate frame for the colour chooser as it needs some space
        JFrame frame2 = new JFrame("Colour Chooser");
        frame2.getContentPane().add(chooser);

        // "register" the window(frame) listener)
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        // register the button listener
        button.addActionListener(new MyActionListener(frame, chooser));

        frame.getContentPane().add(button, BorderLayout.NORTH);

        frame.setSize(800, 800);
        frame.setVisible(true);

        frame2.pack();
        frame2.setVisible(true);
    }
}
```

7 Displaying Images

Just make sure that you have a file with an image in the current directory and that you type the full name of the file (e.g. peppers.png.

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

class ImageViewerExample {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Colour Chooser");
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the filename of the image that " +
                         " you would like to be displayed: ");
        String filename = sc.next();

        frame.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE );
        JLabel label = new JLabel (new ImageIcon (filename));
        JScrollPane jsp = new JScrollPane (label);

        // add label in the frame
        frame.getContentPane().add(jsp);

        frame.pack();
        frame.setVisible(true);
    }
}
```

8 Working with Files

1. Self study. Ask your tutor.

2.
```
1
23
4
```

   As there is no space between numbers 2 and 3, then the number read will be 23.

3. Self study. `nextDouble`, `nextBoolean`, `nextLine`, ...

4.
```
import java.io.*;
import java.util.Scanner;

public class FileExample2 {
    File fp = new File("my_data.txt");

    // do some file writing
    void write() {
        try {
            PrintWriter pw = new PrintWriter(fp);

            pw.println("string1");
            pw.println("string2");
            pw.println("string3");
            pw.close();  // you better maker sure you don't forget this!
        }
        catch (Exception ex) {
            ex.printStackTrace();
        }
```

```java
    }

    // do some file reading
    void read() {
        Scanner sc = null;
        try {
            sc = new Scanner(fp);
            while (sc.hasNext()) {
                String i = sc.next();
                System.out.println(i);
            }
        }
        catch (FileNotFoundException ex) {
            System.err.println("Exception: " + ex);
        }
        finally {
            if (sc != null)
                sc.close();
        }
    }

    public static void main(String[] args) {
        FileExample2 fileTesting = new FileExample2();
        fileTesting.write();
        fileTesting.read();
    }
}
```

5.
```java
import java.io.*;
import java.util.Scanner;

public class FileExample3 {
    File fp = new File("my_data.txt");

    // do some file writing
    void write() {
        try {
            PrintWriter pw = new PrintWriter(fp);

            pw.println("string1");
            pw.println("1");
            pw.println("string2");
            pw.println("2");
            pw.close();  // you better maker sure you don't forget this!
        }
        catch (Exception ex) {
            ex.printStackTrace();
        }
    }
```

```
        // do some file reading
        void read() {
            Scanner sc = null;
            try {
                sc = new Scanner(fp);
                while (sc.hasNext()) {
                    // you must know what the format of the file (order of strings/ints)
                    String s = sc.next();
                    System.out.println(s);

                    int i = sc.nextInt();
                    System.out.println(i);
                }
            }
            catch (FileNotFoundException ex) {
                System.err.println("Exception: " + ex);
            }
            finally {
                if (sc != null)
                    sc.close();
            }
        }

        public static void main(String[] args) {
            FileExample3 fileTesting = new FileExample3();
            fileTesting.write();
            fileTesting.read();
        }
    }
```

9 Serialisation of Objects

The program's output confirms that the data in the `Date` object just before saved are the same with the data in the `Date` object retrieved from the file.