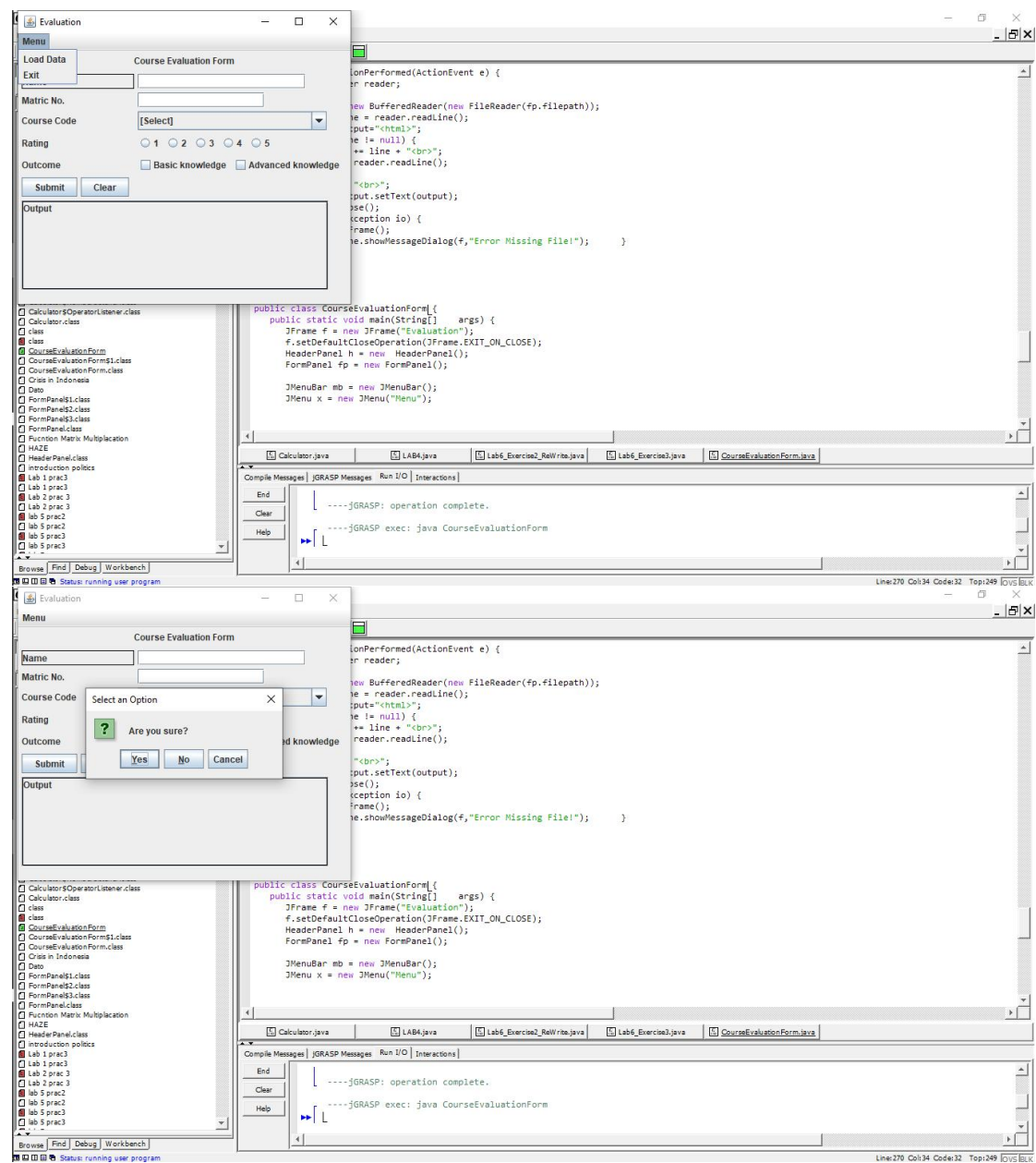


KK14203 LAB 7



Course Evaluation Form

Name: asdasd

Matric No.: asdasd

Course Code: [Select]

Rating: ☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5

Outcome: ☒ Basic knowledge ☐ Advanced knowledge

Submit Clear

Output

Alert: Course Code is not selected

```
public class CourseEvaluationForm {
    public static void main(String[] args) {
        JFrame f = new JFrame("Eval");
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        HeaderPanel h = new HeaderPanel();
        FormPanel fp = new FormPanel();

        JMenuBar mb = new JMenuBar();
        JMenu x = new JMenu("Menu");

        f.add(mb);
        f.add(h);
        f.add(fp);
        f.pack();
        f.setVisible(true);
    }
}
```

Thank you for your evaluation

Name: asdasd
Matric: asdasd
Course: KK14203 Object Oriented Programming
Rating: 5

Message: Data is saved successfully.

Received Files: data.txt C:\Users\User\Documents Open

SOURCE CODE (JAVA)

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;
import javax.swing.BorderFactory;
import javax.swing.border.Border;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.PrintWriter;
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.IOException;

class HeaderPanel extends JPanel{
    private JLabel header;
    public HeaderPanel(){
        header = new JLabel("Course Evaluation Form");
        add(header);
    }
}

class FormPanel extends JPanel implements ActionListener, ItemListener{
    JFrame f;
    JLabel lbl_name;
    JTextField name;
    JLabel lbl_matric;
    JTextField matric;
    JLabel lbl_code;
    JComboBox<String> code;
    JLabel lbl_rating;
    JLabel lbl_outcome;
    JButton b_submit;
    JButton b_clear;
    JCheckBox c1;
    JCheckBox c2;
    JLabel lbl_output;
    JScrollPane jsp;
    Border border = BorderFactory.createLineBorder(Color.BLACK, 1);

    String output="";
    String code_selection="";
    String rb_selection="";
    String cb_selection="";
    String filepath="data.txt";
    String test_IO ="";

    public FormPanel(){
        setLayout(new FlowLayout(FlowLayout.LEFT));

        lbl_name = new JLabel("Name");
        lbl_name.setPreferredSize(new Dimension(150, 20));
        lbl_name.setBorder(border);
```

```

add(lbl_name);
name = new JTextField(20);
add(name);

lbl_matric = new JLabel("Matric No.");
lbl_matric.setPreferredSize(new Dimension(150, 20));
add(lbl_matric);
matric = new JTextField(15);
add(matric);

String[] courses={"[Select]", "KK14203 Object Oriented Programming", "KT14403 Discrete"};

lbl_code = new JLabel("Course Code");
lbl_code.setPreferredSize(new Dimension(150, 20));
add(lbl_code);
code = new JComboBox<String>(courses);
add(code);

code.addActionListener(
    new ActionListener(){
        public void actionPerformed(ActionEvent ae){
            code_selection = (String) code.getSelectedItem();
        }
    });

lbl_rating = new JLabel("Rating");
lbl_rating.setPreferredSize(new Dimension(150, 20));
add(lbl_rating);

JRadioButton rb1 = new JRadioButton("1");
rb1.addActionListener(this);
JRadioButton rb2 = new JRadioButton("2");
rb2.addActionListener(this);
JRadioButton rb3 = new JRadioButton("3");
rb3.addActionListener(this);
JRadioButton rb4 = new JRadioButton("4");
rb4.addActionListener(this);
JRadioButton rb5 = new JRadioButton("5");
rb5.addActionListener(this);
add(rb1);
add(rb2);
add(rb3);
add(rb4);
add(rb5);

ButtonGroup btn_group = new ButtonGroup();
btn_group.add(rb1);
btn_group.add(rb2);
btn_group.add(rb3);
btn_group.add(rb4);
btn_group.add(rb5);

lbl_outcome = new JLabel("Outcome");
lbl_outcome.setPreferredSize(new Dimension(150, 20));
add(lbl_outcome);

c1 = new JCheckBox("Basic knowledge");

```

```

c1.addItemListener(this);
c2 = new JCheckBox("Advanced knowledge");
c2.addItemListener(this);
add(c1);
add(c2);

b_submit = new JButton("Submit");
add(b_submit);
b_clear = new JButton("Clear");
add(b_clear);

b_submit.addActionListener(
    new ActionListener(){
        public void actionPerformed(ActionEvent e){
            f = new JFrame();
            if(name.getText().equals("")){
                JOptionPane.showMessageDialog(f,"Name field is
empty","Alert",JOptionPane.WARNING_MESSAGE);
            }

            else if(matric.getText().equals("")){
                JOptionPane.showMessageDialog(f,"Matric Number is
empty","Alert",JOptionPane.WARNING_MESSAGE);
            }

            else if(checkEmptyString(code_selection) || code_selection.equals("[Select]")){
                JOptionPane.showMessageDialog(null,"Course Code is not
selected","Alert",JOptionPane.WARNING_MESSAGE);}

            else if(checkEmptyString(rb_selection)){
                JOptionPane.showMessageDialog(null,"Rating is not
selected","Alert",JOptionPane.WARNING_MESSAGE);}

            else if(checkEmptyString(cb_selection)){
                JOptionPane.showMessageDialog(null,"Course Outcome is not
selected","Alert",JOptionPane.WARNING_MESSAGE);}

            else{
                if(printOutput()){
                    writeInput();
                    f = new JFrame();
                    JOptionPane.showMessageDialog(f,"Data is saved successfully.");//Show
message if data is saved to text
                }
                else{
                    f = new JFrame();
                    JOptionPane.showMessageDialog(f,"Data saved
unsuccessful","Alert",JOptionPane.WARNING_MESSAGE);}//Show message if data is not save in
                }
            }
        });

b_clear.addActionListener(
    new ActionListener(){
        public void actionPerformed(ActionEvent e){
            lbl_output.setText("Output");
            name.setText("");

```

```

        matric.setText("");
        code.setSelectedIndex(0);
        btn_group.clearSelection();
        c1.setSelected(false);
        c2.setSelected(false);
    }
});

lbl_output = new JLabel("Output");
lbl_output.setBorder(border);
lbl_output.setVerticalAlignment(JLabel.TOP);

jsp = new JScrollPane(lbl_output);
jsp.setPreferredSize(new Dimension(410,120));
add(jsp);
}

public void actionPerformed(ActionEvent ae) {
    rb_selection = ae.getActionCommand();
}

public void itemStateChanged(ItemEvent ie) {
    JCheckBox check = (JCheckBox)ie.getSource();
    cb_selection += check.getText() + " ";
}

public boolean printOutput(){
    output = "<html>";
    output += "Thank you for your evaluation<br><br>";
    output += "Name: " + name.getText() + "<br>";
    output += "Matric: " + matric.getText() + "<br>";
    if(code_selection.equals("[Select]") || code_selection.equals("")){
        lbl_output.setText("Course code is not selected.");
        return false;
    }
    output += "Course: " + code_selection + "<br>";
    output += "Rating: " + rb_selection + "<br>";
    output += "Outcome: " + cb_selection + "<br>";
    output += "</html>";
    lbl_output.setText(output);
    jsp.getViewPort().revalidate();
    return true;
}

public void writeInput(){
    File file = new File(filepath);
    FileWriter fr = null;
    BufferedWriter br = null;
    PrintWriter pr = null;

    String input = name.getText() + ", " + matric.getText() + ", " + code_selection + ", " +
    rb_selection + ", " + cb_selection;

    try {
        fr = new FileWriter(file, true);
        br = new BufferedWriter(fr);
        pr = new PrintWriter(br);
    }
}

```

```

        pr.println(input);
    } catch (IOException e) {
        f = new JFrame();
        JOptionPane.showMessageDialog(f,"Error Missing File!");
    } finally {
        try {
            pr.close();
            br.close();
            fr.close();
        } catch (IOException e) {
            lbl_output.setText(e.toString());
        }
    }
}

public boolean checkEmptyString(String str) {
    if(str != null && !str.isEmpty())
        return false;
    return true;
}

}

class MenuActionListener implements ActionListener {
    FormPanel fp;
    Frame f;
    public MenuActionListener(FormPanel p){
        fp = p;
    }

    public void actionPerformed(ActionEvent e) {
        BufferedReader reader;
        try {
            reader = new BufferedReader(new FileReader(fp.filepath));
            String line = reader.readLine();
            String output="<html>";
            while (line != null) {
                output += line + "<br>";
                line = reader.readLine();
            }
            output += "<br>";
            fp.lbl_output.setText(output);
            reader.close();
        } catch (IOException io) {
            f = new JFrame();
            JOptionPane.showMessageDialog(f,"Error Missing File!");
        }
    }
}

```

```

public class CourseEvaluationForm {
    public static void main(String[] args) {
        JFrame f = new JFrame("Evaluation");
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        HeaderPanel h = new HeaderPanel();
        FormPanel fp = new FormPanel();
    }
}

```

```

JMenuBar mb = new JMenuBar();
JMenu x = new JMenu("Menu");

JMenuItem m1 = new JMenuItem("Load Data");
m1.addActionListener(new MenuActionListener(fp));

JMenuItem m2 = new JMenuItem("Exit");
m2.addActionListener(
    new ActionListener(){
        public void actionPerformed(ActionEvent e){
            int a=JOptionPane.showConfirmDialog(f,"Are you sure?");
            if(a==JOptionPane.YES_OPTION){
                System.exit(0);}}});
x.add(m1);
x.add(m2);

mb.add(x);

f.setJMenuBar(mb);

f.add(h,BorderLayout.NORTH);
f.add(fp, BorderLayout.CENTER);
f.setSize(460,400);
f.setVisible(true);
    }
}

```