

**NAME: LEVANNYAH A/P RAJASEGARAN**

**MATRIC NO.: BI19160337**

**COURSE: KK14203 OBJECT ORIENTED PROGRAMMING (SECTION 1)**

**LECTURER: DR MOHD SHAMRIE SAININ**

## **LAB 7**

1. Add Menu Bar at the top with menu 'Load Data' to read from file and view in the output textarea. Menu 'Exit' to show 'showConfirmDialog' and exit the application if user select 'yes' in the dialog.

```
class MenuActionListener implements ActionListener {
    FormPanel fp;
    //receive FormPanel class to this constructor
    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>";
                // read next line
                line = reader.readLine();
            }
            output += "<br>";
            fp.lbl_output.setText(output);
            reader.close();
        } catch (IOException io) {
            fp.lbl_output.setText(io.toString());
        }
    }
}
```

```
class MenuActionListener2 implements ActionListener{
    FormPanel fp;
    public MenuActionListener2(FormPanel p){
        fp = p;
    }

    //for exit confirmation
    public void actionPerformed(ActionEvent e){
        int confirm = JOptionPane.showConfirmDialog(null, "Are you sure you want to exit?", "Exit Confirmation", JOptionPane.YES_NO_OPTION);
        if (confirm == JOptionPane.YES_OPTION)
            System.exit(0);
    }
}
```

```
// create menuitems
JMenuItem m1 = new JMenuItem("Load Data");
// attach listener and send FormPanel class
m1.addActionListener(new MenuActionListener(fp));

JMenuItem m2 = new JMenuItem("Exit");
m2.addActionListener(new MenuActionListener2(fp));
```

End

Clear

Help

Evaluation

Menu

Load Data

Exit

Course Evaluation Form

Matric No.

Course Code

[Select]

Rating

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Outcome

☐ Basic knowledge ☐ Advanced knowledge

Submit

Clear

Michael, AB191801, KK14203 Object Oriented Programming, 5, Advanced knowledge  
Lee, AC181789, KT14403 Discrete Structures, 4, Advanced knowledge  
Sarah, AD191765, KK14203 Object Oriented Programming, 4, Advanced knowledge  
Jenny, AB181456, KT14403 Discrete Structures, 4, Advanced knowledge

Compile Messages

JGRASP Messages

Run I/O

Interactions

End

Clear

Help

Evaluation

Menu

Course Evaluation Form

Name

Ali

Matric No.

AB181234

Course Code

KT14403 Discrete Structures

Rating

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5

Outcome

☐ Basic knowledge ☐ Advanced knowledge

Submit

Clear

Output

Exit Confirmation

Are you sure you want to exit?

Yes

No

2. Input validation from all input to check if user empty field or selections when user click 'Submit' button.

```
public boolean printOutput(){
    output = "<html>";
    output += "Thank you for your evaluation<br><br>";

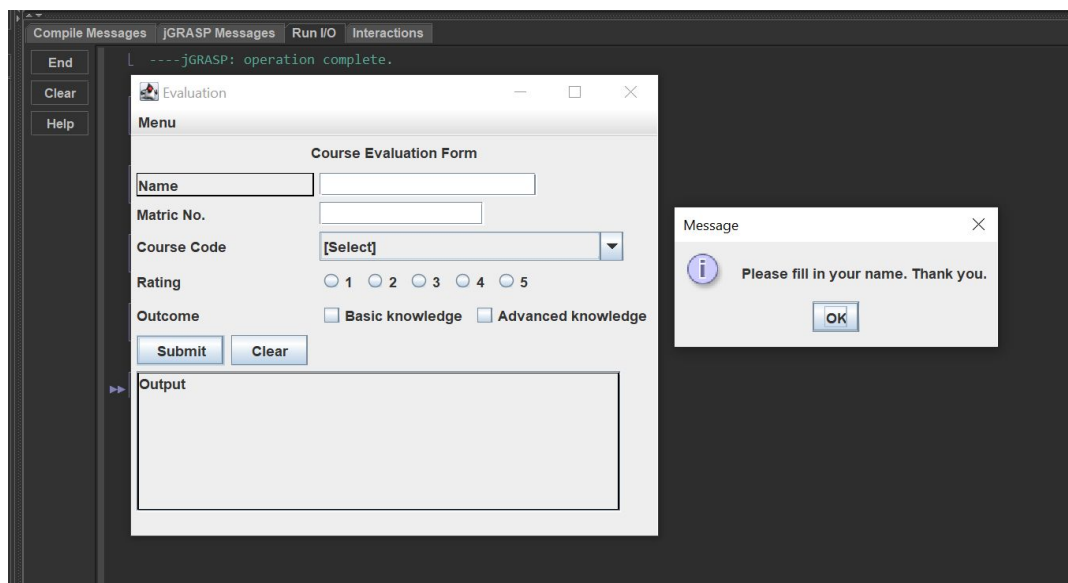
    if(name.getText().isEmpty()){
        JOptionPane.showMessageDialog(null, "Please fill in your name. Thank you.");
        return false;
    }
    output += "Name: " + name.getText() + "<br>";

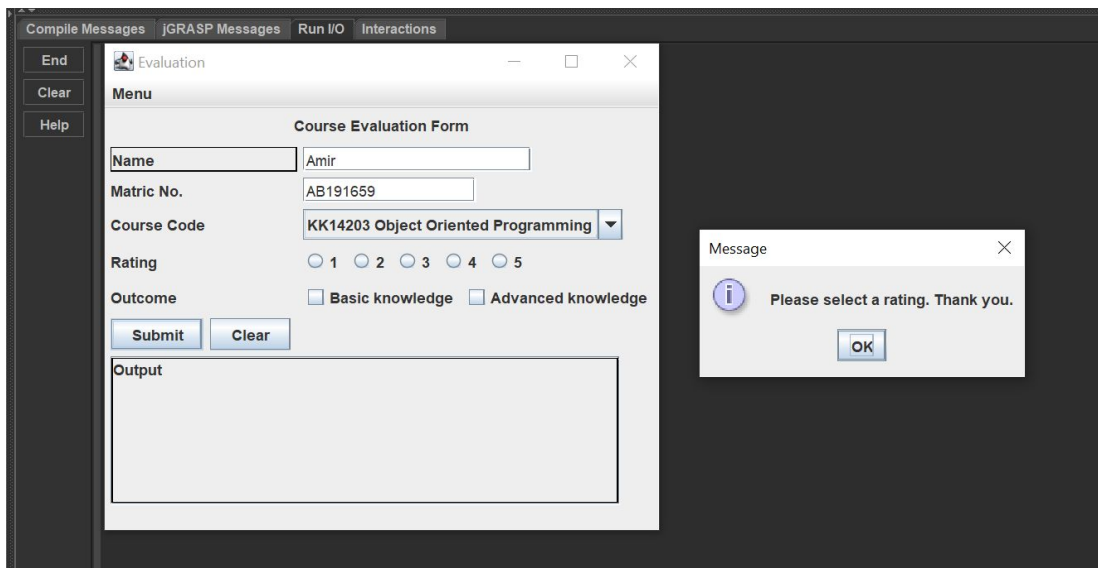
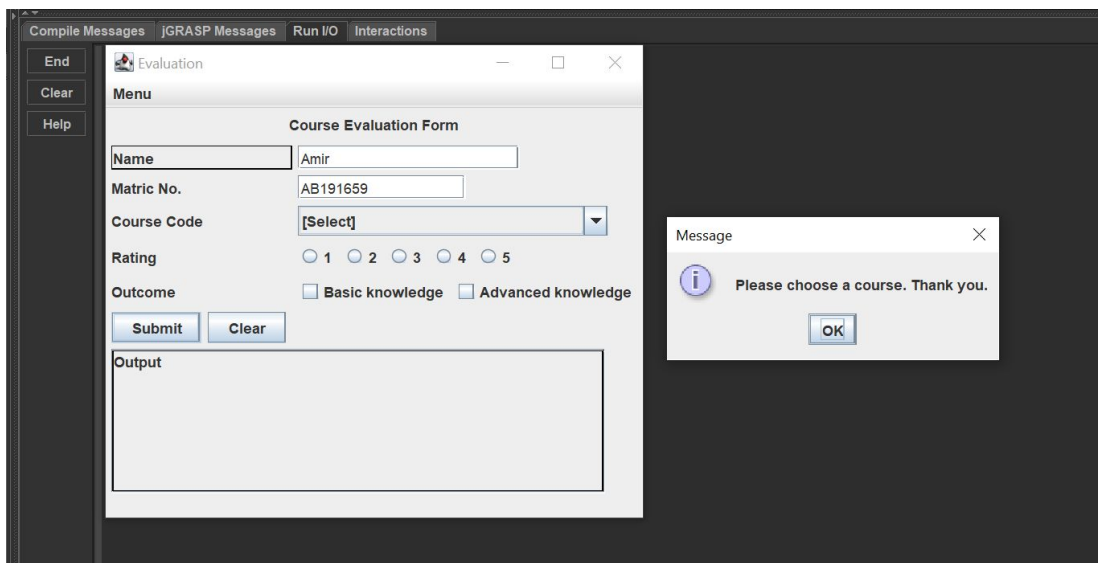
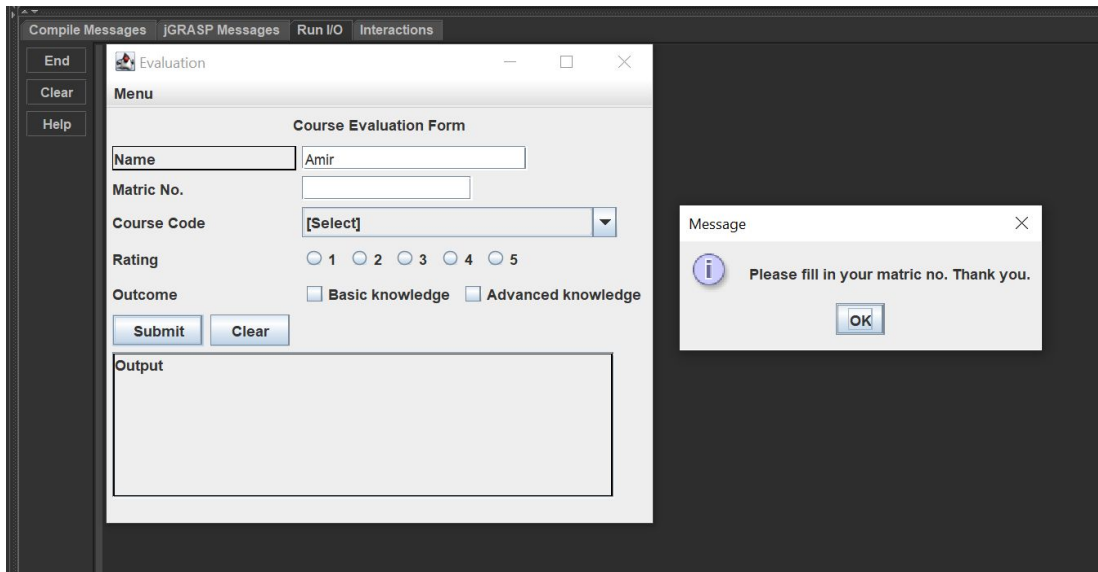
    if(matric.getText().isEmpty()){
        JOptionPane.showMessageDialog(null, "Please fill in your matric no. Thank you.");
        return false;
    }
    output += "Matric: " + matric.getText() + "<br>";

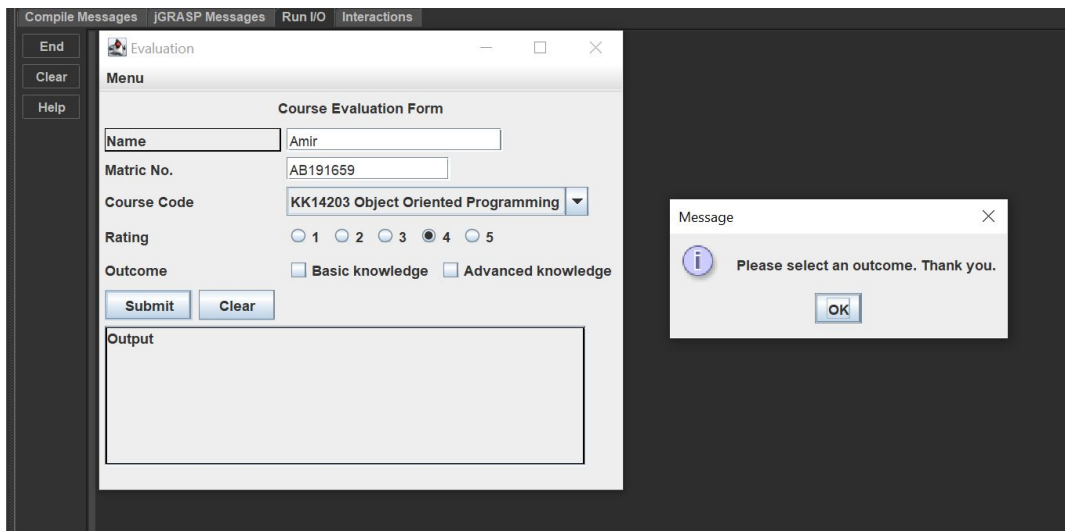
    if(code_selection.equals("[Select]") || code_selection.equals(" ")){
        JOptionPane.showMessageDialog(null, "Please choose a course. Thank you.");
        return false;
    }
    output += "Course: " + code_selection + "<br>";

    if(rb_selection.equals(" ")){
        JOptionPane.showMessageDialog(null, "Please select a rating. Thank you.");
        return false;
    }
    output += "Rating: " + rb_selection + "<br>";

    if(cb_selection.equals(" ")){
        JOptionPane.showMessageDialog(null, "Please select an outcome. Thank you.");
        return false;
    }
    output += "Outcome: " + cb_selection + "<br>";
    output += "</html>";
    lbl_output.setText(output);
    jsp.getViewport().revalidate();
    return true;
}
```





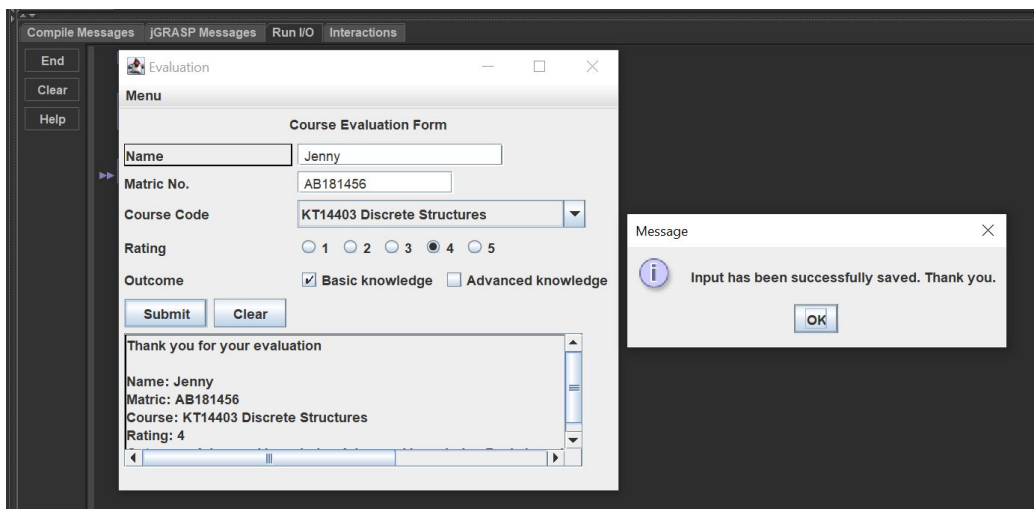


3. Save (add) the data into a text file with dialog notification (e.g. showMessageDialog) whether input is successfully saved.

```
//write to file
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;

    //exception implementation
    try {
        // to append to file, you need to initialize FileWriter using below constructor
        fr = new FileWriter(file, true);
        br = new BufferedWriter(fr);
        pr = new PrintWriter(br);
        pr.println(input);
        JOptionPane.showMessageDialog(null, "Input has been successfully saved. Thank you.");
    } catch (IOException e) {
        lbl_output.setText(e.toString());
        JOptionPane.showMessageDialog(null, "Something went wrong. Please try again.");
    } finally {
        try {
            pr.close();
            br.close();
            fr.close();
        } catch (IOException e) {
            lbl_output.setText(e.toString());
        }
    }
}
}
```



4. Implement at least ONE (1) exception handling (e.g. file IO and dealing with empty input field).

```
//exception implementation
try {
    // to append to file, you need to initialize FileWriter using below constructor
    fr = new FileWriter(file, true);
    br = new BufferedWriter(fr);
    pr = new PrintWriter(br);
    pr.println(input);
    JOptionPane.showMessageDialog(null, "Input has been successfully saved. Thank you.");
} catch (IOException e) {
    lbl_output.setText(e.toString());
    JOptionPane.showMessageDialog(null, "Something went wrong. Please try again.");
} finally {
    try {
        pr.close();
        br.close();
        fr.close();
    } catch (IOException e) {
        lbl_output.setText(e.toString());
    }
}
}
```

```
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>";
            // read next line
            line = reader.readLine();
        }
        output += "<br>";
        fp.lbl_output.setText(output);
        reader.close();
    } catch (IOException io) {
        fp.lbl_output.setText(io.toString());
    }
}
```