



UMS
UNIVERSITI MALAYSIA SABAH

FACULTY OF COMPUTING AND INFORMATICS

HC 00 SOFTWARE ENGINEERING

SEMESTER 2 (2019 / 2020)

KK 14203 OBJECT ORIENTED PROGRAMMING

SECTION 1

INDIVIDUAL ASSIGNMENT

LAB 7 : GUI

PREPARED BY

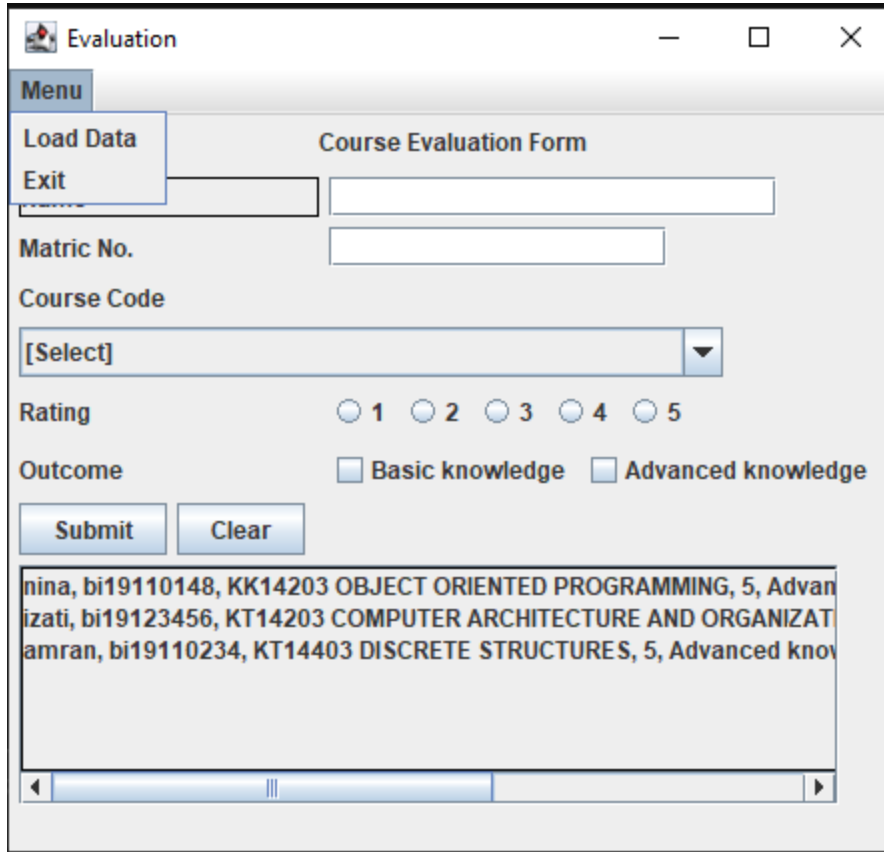
AMANINA IZATI BINTI AMRAN (BI19110148)

PREPARED FOR

DR. MOHD SHAMRIE SAININ

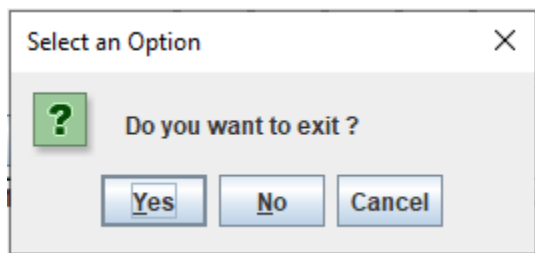
Requirements :

1. Add Menu Bar at the top with menu 'Load Data' to read from file and view in the output textrea.



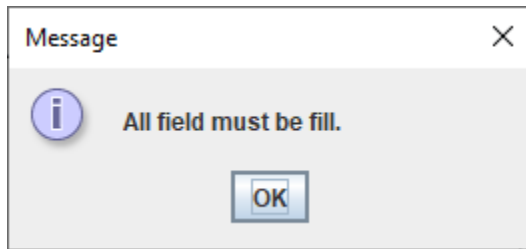
The screenshot shows a Java Swing window titled "Evaluation" with a standard Mac OS-style title bar (minimize, maximize, close buttons). Inside the window, there is a menu bar with a "Menu" dropdown menu. The menu contains two items: "Load Data" and "Exit". Below the menu bar, the window is titled "Course Evaluation Form". It contains several input fields: a text field for "Matric No.", a text field for "Course Code", and a dropdown menu with "[Select]" as the current selection. Below these fields are five radio buttons labeled "1", "2", "3", "4", and "5" for the "Rating". There are also two checkboxes labeled "Basic knowledge" and "Advanced knowledge" for the "Outcome". At the bottom left, there are two buttons: "Submit" and "Clear". At the bottom right, there is a large text area containing the following text: "nina, bi19110148, KK14203 OBJECT ORIENTED PROGRAMMING, 5, Advan", "izati, bi19123456, KT14203 COMPUTER ARCHITECTURE AND ORGANIZAT", "amran, bi19110234, KT14403 DISCRETE STRUCTURES, 5, Advanced know". The text area has a scrollbar on the right side.

2. Menu 'Exit' to show 'showConfirmDialog' and exit the application if user select 'yes' in the dialog.

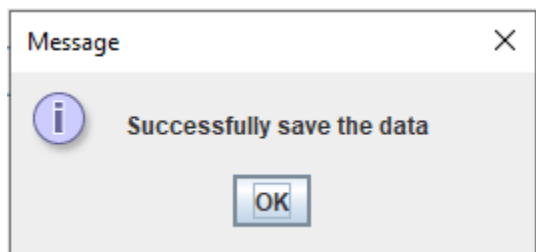


The screenshot shows a Java Swing dialog box titled "Select an Option" with a close button (X) in the top right corner. Inside the dialog, there is a green square icon with a white question mark. To the right of the icon, the text "Do you want to exit ?" is displayed. At the bottom of the dialog, there are three buttons: "Yes", "No", and "Cancel".

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



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



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

```
try {  
    fr = new FileWriter(file, true);  
    br = new BufferedWriter(fr);  
    pr = new PrintWriter(br);  
    pr.println(input);  
    JOptionPane.showMessageDialog(null, "Successfully save the data.");  
} catch (IOException e) {  
    lbl_output.setText(e.toString());  
    JOptionPane.showMessageDialog(null, "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>";
            line = reader.readLine();
        }
        output += "<br>";
        fp.lbl_output.setText(output);
        reader.close();
    }
    catch (IOException io)
    {
        fp.lbl_output.setText(io.toString());
    }
}
}
```