

CRYSTAL MARVEENA MARITIN (BI19110079)

Requirements:

1. Add Menu Bar at the top with menu 'Load Data' to read from file and view in the output text area. (figure 1.1) Menu 'Exit' to show 'showConfirmDialog' and exit the application if user select 'yes' in the dialog. (figure 1.2)

Course Evaluation

Menu

Load Data

Exit

COURSE EVALUATION FORM

Name: CRYSTAL MARVEENA MARITIN

Matric No.: BI19110079

Course Code: KK14203 Oriented Object Programming

Rating: 1 2 3 4 5

OUTCOME: ☐ BASIC KNOWLEDGE ☒ ADVANCED KNOWLEDGE

SUBMIT CLEAR

CRYSTAL MARVEENA MARITIN || BI19110079 || KK14203 Oriented Obj
CRYSTAL MARVEENA MARITIN || BI19110079 || KK14203 Oriented Obj
CRYSTAL MARVEENA MARITIN || BI19110079 || KK14203 Oriented Obj

FIGURE 1.1 Menu Bar at the top with menu 'Load Data' to read from file and view in the output text area.

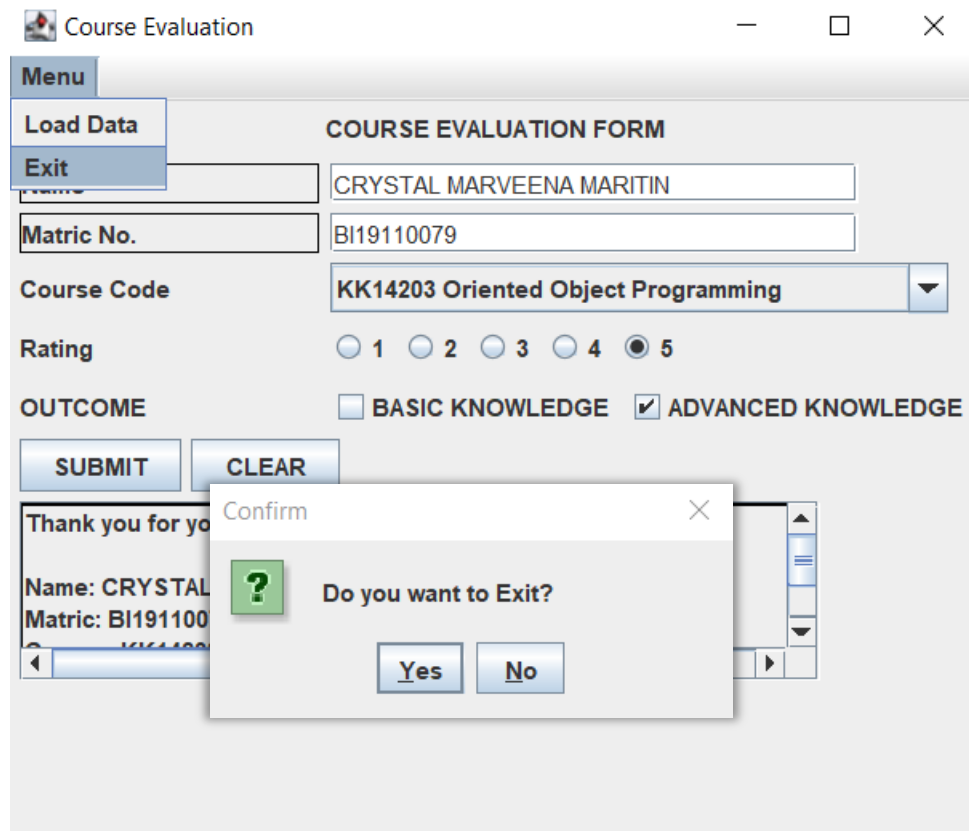
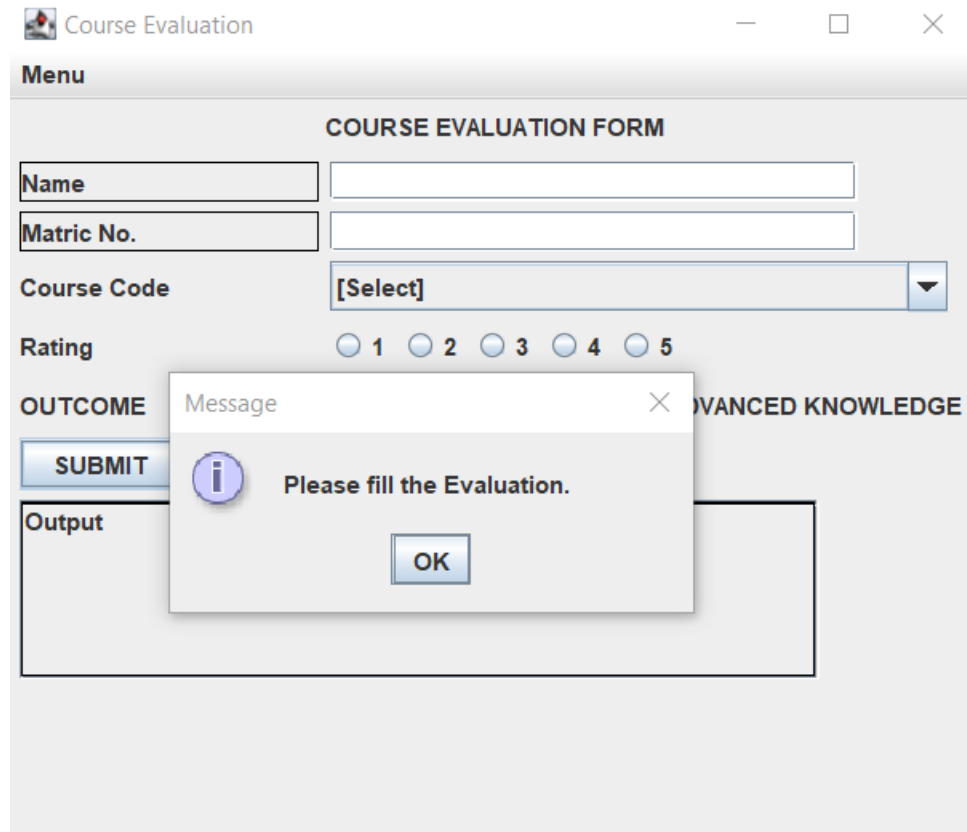


FIGURE 1.2 Menu 'Exit' to show 'showConfirmDialog' and to exit the application if user click 'Yes' in the dialog

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



The image shows a Java Swing window titled "Course Evaluation". Inside the window, there is a "Menu" bar and a "COURSE EVALUATION FORM". The form contains several input fields: "Name", "Matric No.", and "Course Code" (a dropdown menu currently showing "[Select]"). Below these is a "Rating" section with five radio buttons labeled 1 through 5. To the left of the form, there is a "SUBMIT" button. A modal "Message" dialog box is displayed over the form, containing an information icon, the text "Please fill the Evaluation.", and an "OK" button. The dialog box is positioned over the "SUBMIT" button and the "Output" label. The "Output" label is located below the "SUBMIT" button. The "COURSE EVALUATION FORM" also includes a section labeled "OUTCOME" and "ADVANCED KNOWLEDGE" which are partially visible.

FIGURE 2.1

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

The screenshot shows a Java Swing window titled "Course Evaluation". Inside the window, there is a section titled "COURSE EVALUATION FORM". This section contains several input fields: "Name" with the value "CRYSTAL MARVEENA MARITIN", "Matric No." with the value "BI19110079", and "Course Code" with a dropdown menu showing "KK14203 Oriented Object Programming". Below these fields is a "Rating" section with five radio buttons labeled 1 through 5; the radio button for "5" is selected. To the left of the "Rating" section is a "Menu" section with a "SUBMIT" button. A "Message" dialog box is overlaid on the "SUBMIT" button, displaying an information icon, the text "Data saved successfully", and an "OK" button. Below the "SUBMIT" button, there is a "Thank you for" message and a list of names, including "Name: CRYSTAL MARVEENA MARITIN" and "Matric: BI19110079".

FIGURE 3.1

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

```
//exception implementation - to append new data into text file
try {
    // FileWriter is initialized by using the following constructors

    fr = new FileWriter(file, true);
    br = new BufferedWriter(fr);
    pr = new PrintWriter(br);
    pr.println(input);
}

catch (IOException e) {
    output_label.setText(e.toString());
} finally {
    try {
        pr.close();
        br.close();
        fr.close();
    } catch (IOException e) {
        output_label.setText(e.toString());
    }
}
```

FIGURE 4.1 EXCEPTION HANDLING OF APPENDING NEW DATA INTO EXISTING DATA TEXT FILE