



UMS
UNIVERSITI MALAYSIA SABAH

KK14203

PENGATURCARAAN BERORIENTASIKAN OBJEK

SECTION 1

INDIVIDUAL TASK

LAB 7

DR. SAMRY @ MOHD. SHAMRIE SAININ

STUDENT NAME

FRANCES LYNNE WILLIAM

MATRIC NUM.

BI19160328

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.

The screenshot shows a Java Swing window titled "Evaluation". It features a menu bar with "Menu" selected, which has opened a dropdown menu containing "Load Data" and "Exit". The main area is titled "Course Evaluation Form" and contains several input fields: a text field for "Name", a text field for "Matric No.", a dropdown menu for "Course Code" with "[Select]" as the current value, and a radio button group for "Rating" with options 1, 2, 3, 4, and 5. Below these is a checkbox group for "Outcome" with options "Basic knowledge" and "Advanced knowledge". At the bottom left are "Submit" and "Clear" buttons. A large text area labeled "Output" is at the bottom right.

This screenshot shows the same "Evaluation" application window, but with a confirmation dialog box open over the "Exit" menu option. The dialog box is titled "Select an option" and contains a green question mark icon, the text "Are you sure you want to exit?", and two buttons labeled "Yes" and "No". The background application window is slightly dimmed.

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 "Evaluation" with a standard Mac OS-style title bar (minimize, maximize, close buttons). The window contains a "Course Evaluation Form" with the following components:

- Menu**: A label at the top left of the form area.
- Form Fields**:
 - Name**: A text input field.
 - Matric No.**: A text input field.
 - Course Code**: A text input field.
 - Rating**: A text input field.
 - Outcome**: A text input field.
- Buttons**:
 - Submit**: A button located below the Outcome field.
 - Clear**: A button located to the right of the Submit button.
- Output**: A large text area at the bottom of the form.

A "Message" dialog box is currently open, displaying an information icon (i) and the text "Please fill out all field!". It has an "OK" button at the bottom.

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 "Evaluation". Inside, there is a "Course Evaluation Form" with the following fields: "Name" (containing "Frances"), "Matric No." (containing "BI19160328"), "Course Code" (containing "KP14203 OOP"), "Rating" (containing "5"), and "Outcome" (containing "Basic knowledge"). Below the form is an "Output" area displaying: "Thank you for your evaluation", "Name: Frances", "Matric: BI19160328", "Course: KP14203 OOP", "Rating: 5", and "Outcome: Basic knowledge". A "Message" dialog box is overlaid on the form, displaying an information icon and the text: "Congratulations! You have successfully submit the course evaluation form for KP14203 OOP !". The dialog has an "OK" button. A red bracket on the right side of the dialog points to the text "messageDialog)".

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

The screenshot shows the same "Evaluation" window, but the form fields are empty: "Name", "Matric No.", "Course Code" (with a "[Select]" dropdown), "Rating", and "Outcome". Below the form are "Submit" and "Clear" buttons. An "Output" area is visible at the bottom. A "Message" dialog box is overlaid on the form, displaying an information icon and the text: "Please fill out all field!". The dialog has an "OK" button.