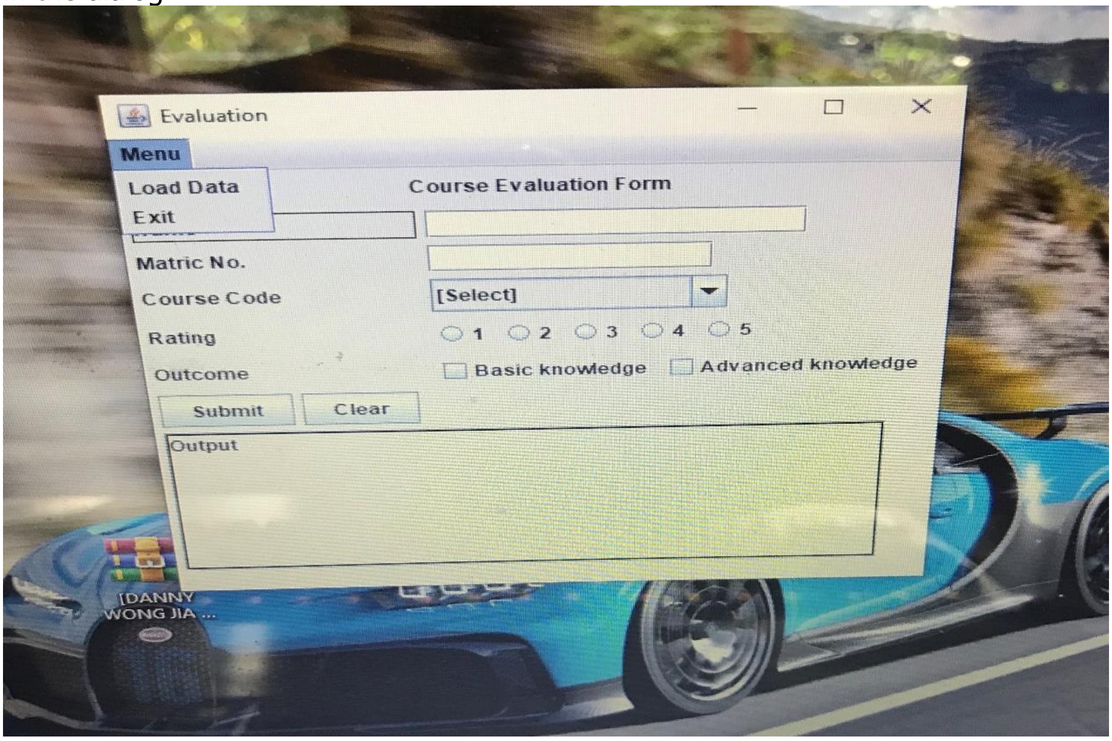
LAB 7

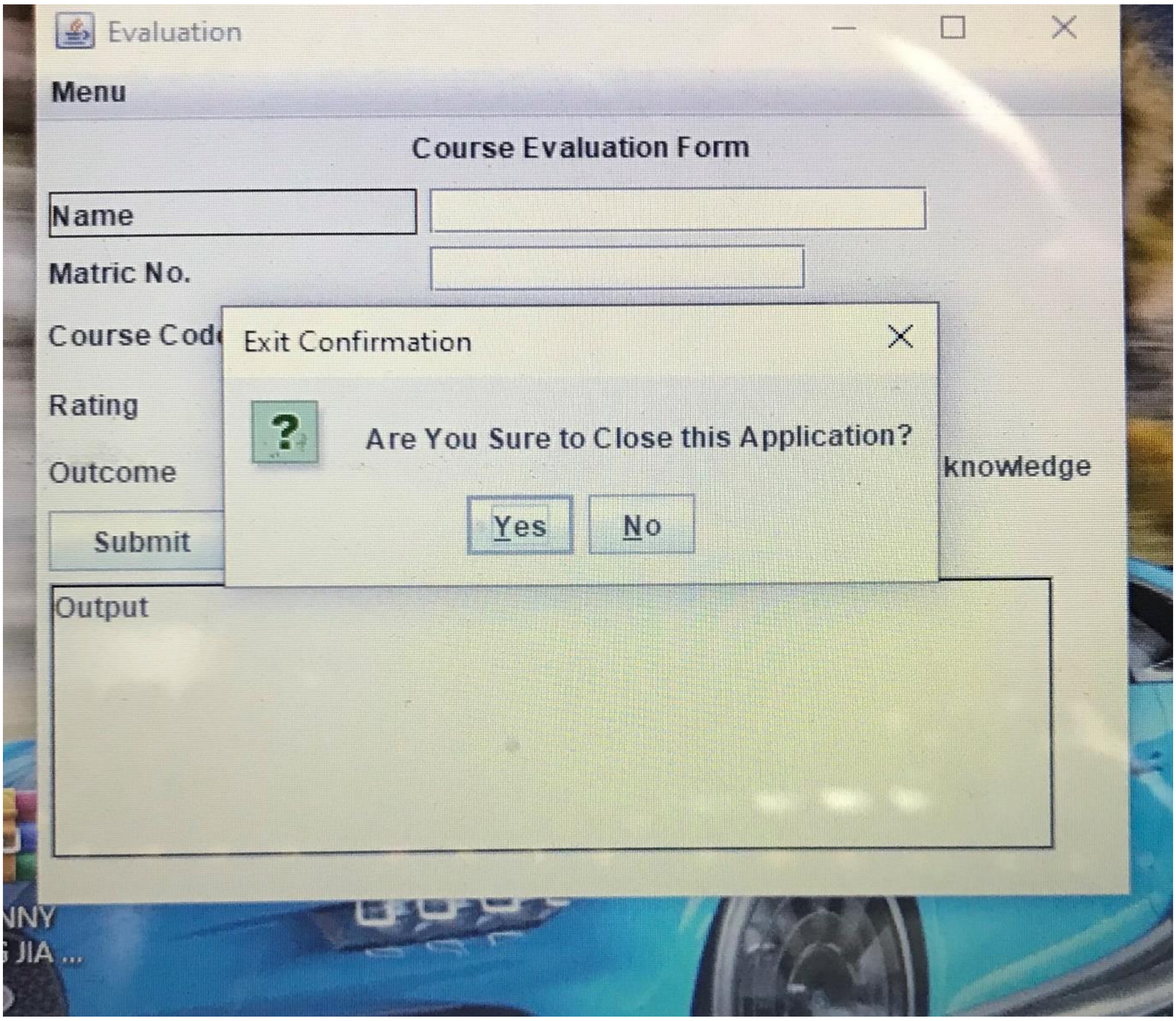
NAME:DANNY WONG JIA HONG

MATRICS NO :BI19110060

SECTION : 1

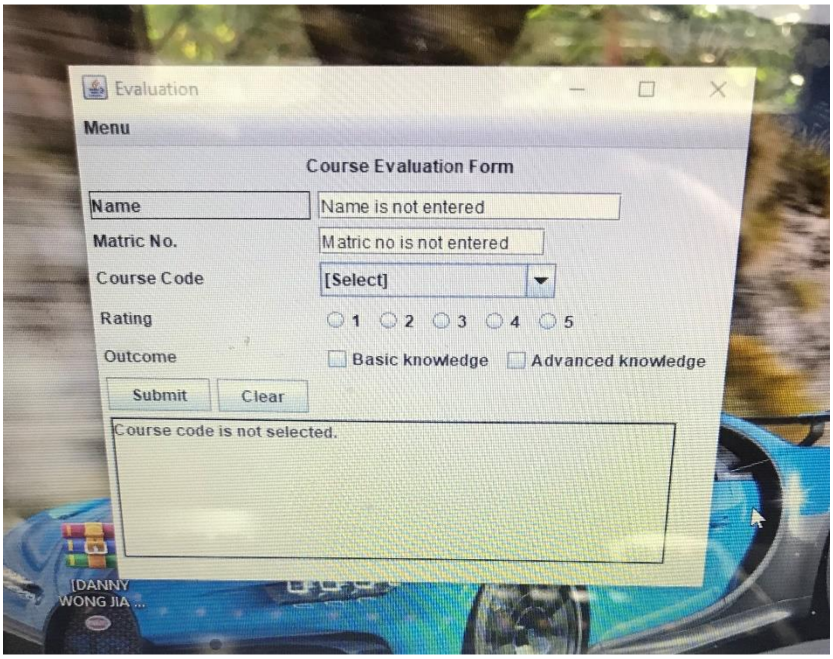
1. Add Menu Bar at the top with menu 'Load Data' to read from file and view in the output textrea. Menu 'Exit' to show 'showConfirmDialog' and exit the application if user select 'yes' in the dialog.





Are You Sure to Close this Application?

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



Course Evaluation Form

Name: Name is not entered

Matric No.: Matric no is not entered

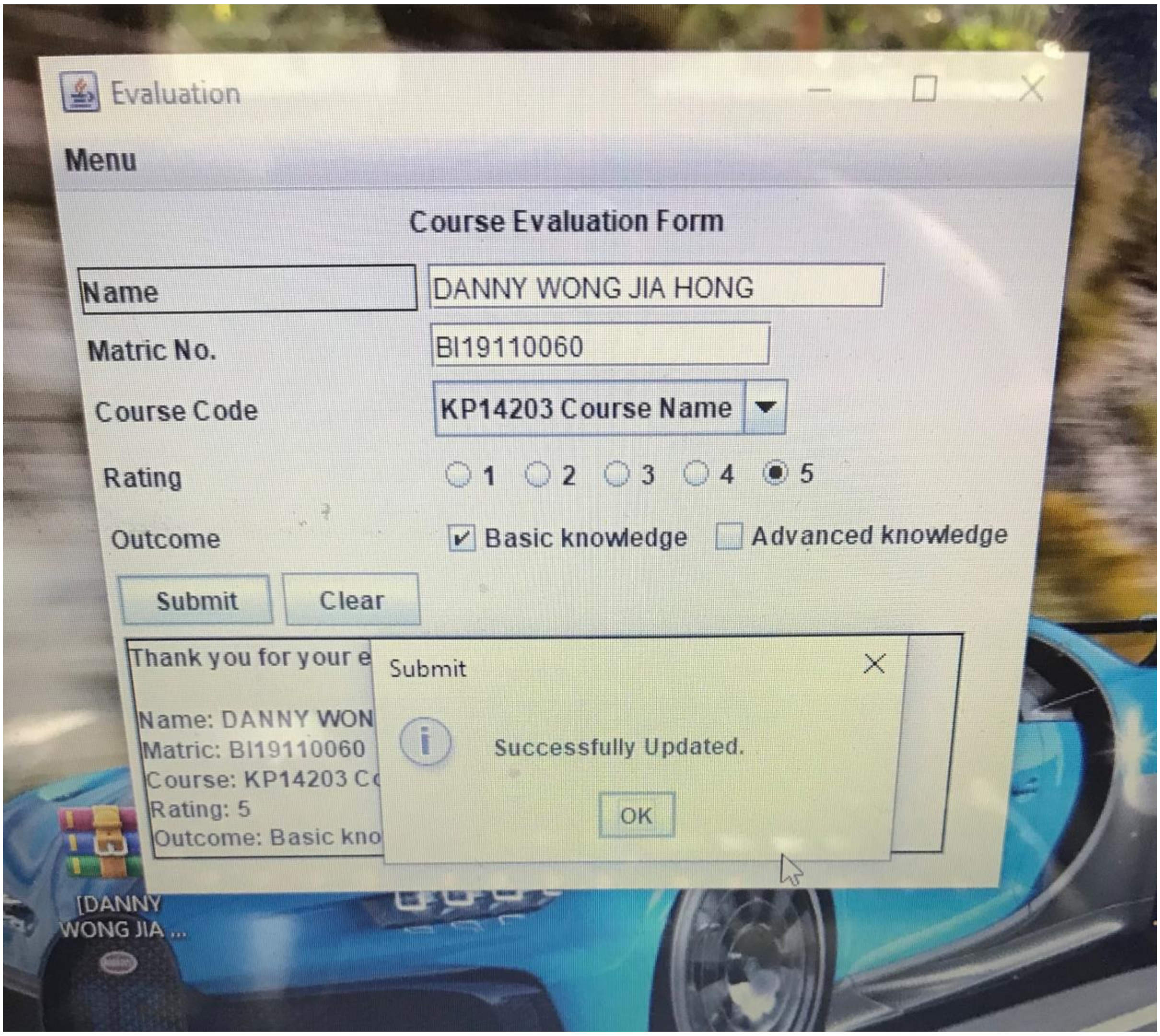
Course Code: [Select]

Rating: ○ 1 ○ 2 ○ 3 ○ 4 ○ 5

Outcome: □ Basic knowledge □ Advanced knowledge

Course code is not selected.

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



Course Evaluation Form

Name: DANNY WONG JIA HONG

Matric No.: BI19110060

Course Code: KP14203 Course Name

Rating: ○ 1 ○ 2 ○ 3 ○ 4 ● 5

Outcome: ☑ Basic knowledge □ Advanced knowledge

Thank you for your e

Name: DANNY WON
Matric: BI19110060
Course: KP14203 Co
Rating: 5
Outcome: Basic kno

Submit

Successfully Updated.

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

Code

```
import java.awt.*;

import java.awt.event.*;


//required for border

import javax.swing.*;

import javax.swing.event.*;

import javax.swing.border.Border;

//required for file IO

import java.io.File;

import java.io.FileReader;

import java.io.FileWriter;

import java.io.PrintWriter;

import java.io.BufferedWriter;

import java.io.BufferedReader;

//required for exception

import java.io.IOException;



//Header panel

class HeaderPanel extends JPanel{

  private JLabel header;

  public HeaderPanel(){

            header = new JLabel("Course Evaluation Form");

            add(header);

  }

}


//Form Panel

class FormPanel extends JPanel implements ActionListener, ItemListener {

  //list all UI components for the panel

  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);

//global variable
String output="";
String code_selection="";
String rb_selection="";
String cb_selection="";
String filePath="data.txt"; //in the same directory



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]", "KP14203 Course Name", "KT20363 Course Name"};

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

//JComboBox action listener
code.addActionListener(new ActionListener(){
  public void actionPerformed(ActionEvent ae){
    //get selected item
    code_selection = (String) code.getSelectedItem();
  }
});

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

//Radio buttons and action listener
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);

 //define button group
 ButtonGroup bg = new ButtonGroup();
 bg.add(rb1);
 bg.add(rb2);
 bg.add(rb3);
 bg.add(rb4);
 bg.add(rb5);

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

 //checkbox and  item listener
 c1 = new JCheckBox("Basic knowledge");
 c1.addItemListener(this);
 c2 = new JCheckBox("Advanced knowledge");
 c2.addItemListener(this);
 add(c1);
 add(c2);
 JFrame f=new JFrame();
b_submit = new JButton("Submit");
 add(b_submit);
 b_clear = new JButton("Clear");
 add(b_clear);

 //handle button submit action listener
 //view the input to output label
 //and write to file
```

```
b_submit.addActionListener(new ActionListener(){
  public void actionPerformed(ActionEvent e){

    //call method

          if(e.getSource()== b_submit && printOutput()) {

                      JOptionPane.showMessageDialog(f,"Successfully Updated.","Submit",JOptionPane.NO_OPTION);
                      writeInput();
          }

  //   if(printOutput())
  //     writeInput();



  }
});

//handle button clear action listener
b_clear.addActionListener(new ActionListener(){
  public void actionPerformed(ActionEvent e){
    lbl_output.setText("Output");
    name.setText("");
    matric.setText("");
    code.setSelectedIndex(0);
    bg.clearSelection();
    c1.setSelected(false);
    c2.setSelected(false);
  }
});

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

```
  //add output label to scrollpane
  jsp = new JScrollPane(lbl_output);
  jsp.setPreferredSize(new Dimension(410,120));
  add(jsp);
}

//handle radio button selection
public void actionPerformed(ActionEvent ae) {
  rb_selection = ae.getActionCommand();
}

//handle item listener for checkbox
public void itemStateChanged(ItemEvent ie) {
  JCheckBox check = (JCheckBox)ie.getSource();
  cb_selection += check.getText() + " ";
}

//method to print output to lbl_output
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("")){
          name.setText("Name is not entered");
          matric.setText("Matric no is not entered");
   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;
  }

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

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());
                        }

 }
}




//run the application using this main
public class CourseEvaluationApp extends JFrame {
            public static JFrame f = new JFrame("Evaluation");
            public static JMenuBar mb = new JMenuBar();
            // create a menu
            public static JMenu x = new JMenu("Menu");
```

```
// create menuitems
        public static JMenuItem m1 = new JMenuItem("Load Data");
        public static JMenuItem Exit = new JMenuItem("Exit");



         public CourseEvaluationApp() {

             x.add(Exit);
             mb.add(x);
             Exit.addActionListener(new ExitListener());
             WindowListener exitListener = new WindowAdapter() {

               @Override
               public void windowClosing(WindowEvent e) {
                 int confirm = JOptionPane.showOptionDialog(f,
                     "Are You Sure to Close this Application?",
                     "Exit Confirmation", JOptionPane.YES_NO_OPTION,
                     JOptionPane.QUESTION_MESSAGE, null, null, null);
                 if (confirm == JOptionPane.YES_OPTION) {
                   System.exit(1);
                 }
               }
             };
             f.addWindowListener(exitListener);
             f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
             f.setJMenuBar(mb);
             f.setPreferredSize(new Dimension(400, 300));
             f.setLocation(100, 100);
             f.pack();
             f.setVisible(true);
          }
         class ExitListener implements ActionListener {

             @Override
             public void actionPerformed(ActionEvent e) {
                    int confirm = JOptionPane.showOptionDialog(f,
                        "Are You Sure to Close this Application?",
                        "Exit Confirmation", JOptionPane.YES_NO_OPTION,
                        JOptionPane.QUESTION_MESSAGE, null, null, null);

                    if (confirm == JOptionPane.YES_OPTION) {
                      System.exit(1);
                    }
                  }
                }


  public static void main(String[] args) {



//     f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    //load panels
    HeaderPanel h = new        HeaderPanel();
   HeaderPanel h2 = new       HeaderPanel();
    FormPanel fp = new FormPanel();

    m1.addActionListener(new MenuActionListener(fp));




//    JMenuItem Exit = new JMenuItem("Exit");
  //  m2.addActionListener((event) -> System.exit(0));
    CourseEvaluationApp cea = new CourseEvaluationApp();



    // add menu items to menu
    x.add(m1);
    x.add(Exit);

    // add menu to menu bar
```

```
     mb.add(x);
     // add menubar to frame
     f.setJMenuBar(mb);


     //add panels to frame
     f.add(h,BorderLayout.NORTH);
     f.add(fp, BorderLayout.CENTER);
     f.setSize(460,400);
     f.setVisible(true);
   }
}
```