Lab Exercise for Week 11

JAVA GRAPHICAL USER INTERFACE

BITP 3113 Object-Oriented Programming

Author

Dr. Emaliana Kasmuri

Fakulti Teknologi Maklumat dan Komunikasi

Universiti Teknikal Malaysia Melaka

Table of Contents

Learning Outcomes

At the end of this lab exercise, the student should be able to: -

1. Create an empty Java window.
2. Create window-based applications using Java graphical user interface components.

Tools

The exercise for this lab session is using the following tools: -

1. Eclipse for Java Developers

Supporting Materials

Java API documentation at
https://docs.oracle.com/en/java/javase/22/docs/api/index.html.

Section 1: Java Window Frame Development

This section describes exercises to develop a Java window frame.

Exercise 1: First Java Frame Definition

The following steps create a new Java frame.

a. Create a new Java class named `FirstFrame`. This class will extend to `JFrame`.
b. Define a private method named `setWindowProperties()`. This method will not receive any parameters and return any value. The purpose of this method is to define the window properties.
c. Provide the implementation of the method using Java code in Figure 1. Observe the code as you type.

```java
// Set the title of the window
super.setTitle("My First Java GUI");

// Set the size of the window: width = 400, Height = 300
super.setSize(400, 300);

// Set the default close operation
super.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

// Center the window on the screen
super.setLocationRelativeTo(null);

// Make the window visible
super.setVisible(true);
```

Figure 1: Java code for the method setWindowProperties( ) implementation

d. Invoke this method from the class's constructor.
e. Provide suitable comments for the class and the method.
f. Import the GUI component class from `javax.swing` package.
g. Fix any errors.
h. Save the class.

Exercise 2: Java Application Definition

a. Create a new class with a `main( )` method named `FirstFrameDemo`.
b. Create an object from the class defined in Exercise 1.
c. Provide suitable comments for the class and the method.
d. Fix any errors.
e. Save the class.
f. Execute the application. The output shall be similar as shown in Figure 2.

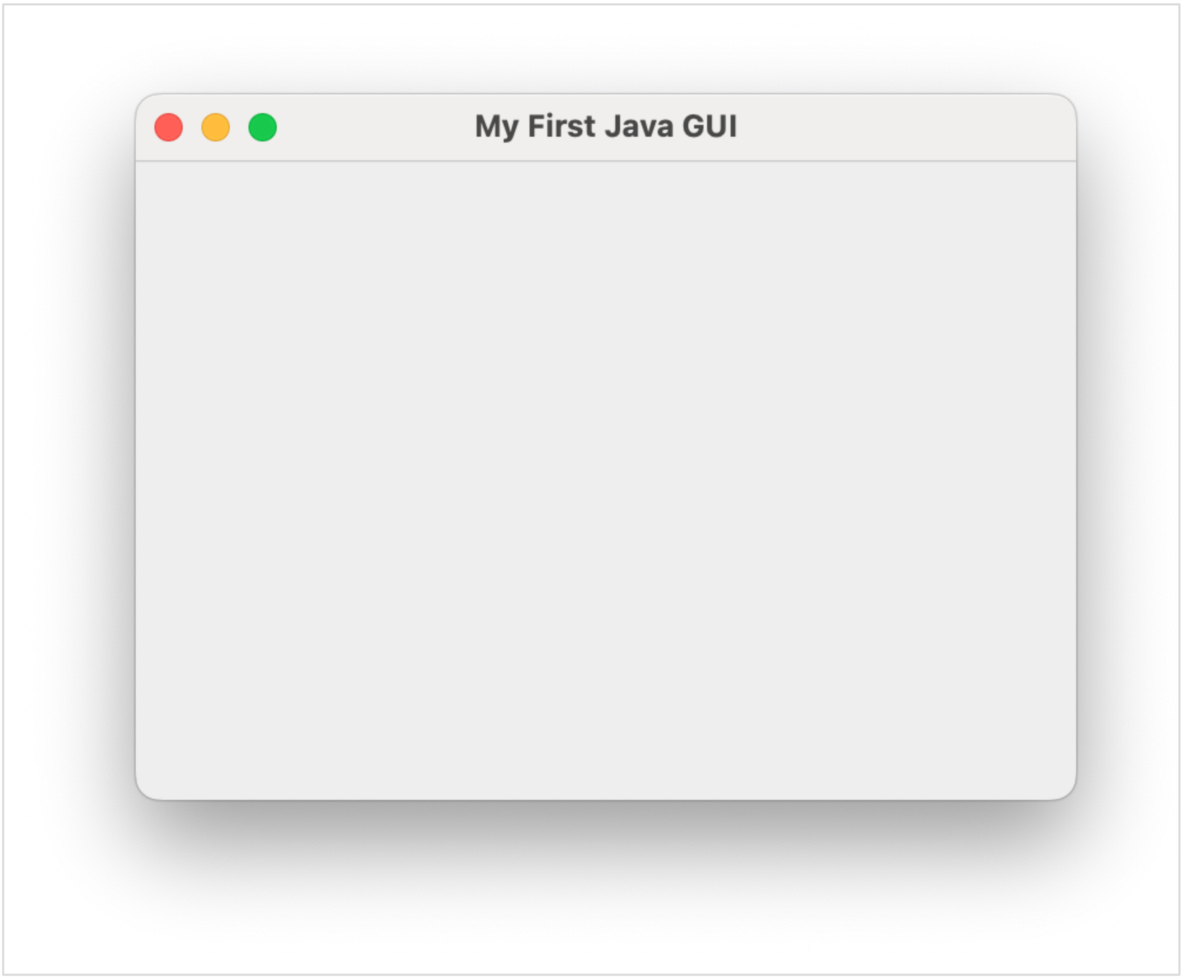

Figure 2: The expected output from Exercise 2

g. Observe the window behaviors as follows: -
    a. Move the window around the screen.
    b. Minimize and maximize the window using the available buttons.
h. Click the suitable button to close the window.

Exercise 3: Resizing the Window

a. Open **the FirstFrame.java** file.
b. Implement the requirements shown in Table 1.

Table 1: New window requirement for class FirstFrame

Property	Value
Size	500x400
Title	Demo Application

c. Fix any errors.
d. Save the class.
e. Execute class `FirstFrameDemo`.
f. Observe the changes.

Exercise 4: Java JFrame comprehension

Answer the following questions using the knowledge and skill comprehended from the previous exercises. Record your answer in ulearn.

a. Describe how should a programmer create a Java desktop screen? Use example to support your answer.
b. State the method that allow the screen to be visible on the screen.
c. Describe the screen behavior when the command in Figure 3 is invoked.

```
super.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

Figure 3: Java method

Section 2: Understanding Common Java GUI Components

a. Download the **Week 11 - Section 02 - Supplementary Codes.zip** file from ulearn.
b. Unzip the file.
c. Execute the MainGUIApp.java. The initial output shall be similar as shown in Figure 1.

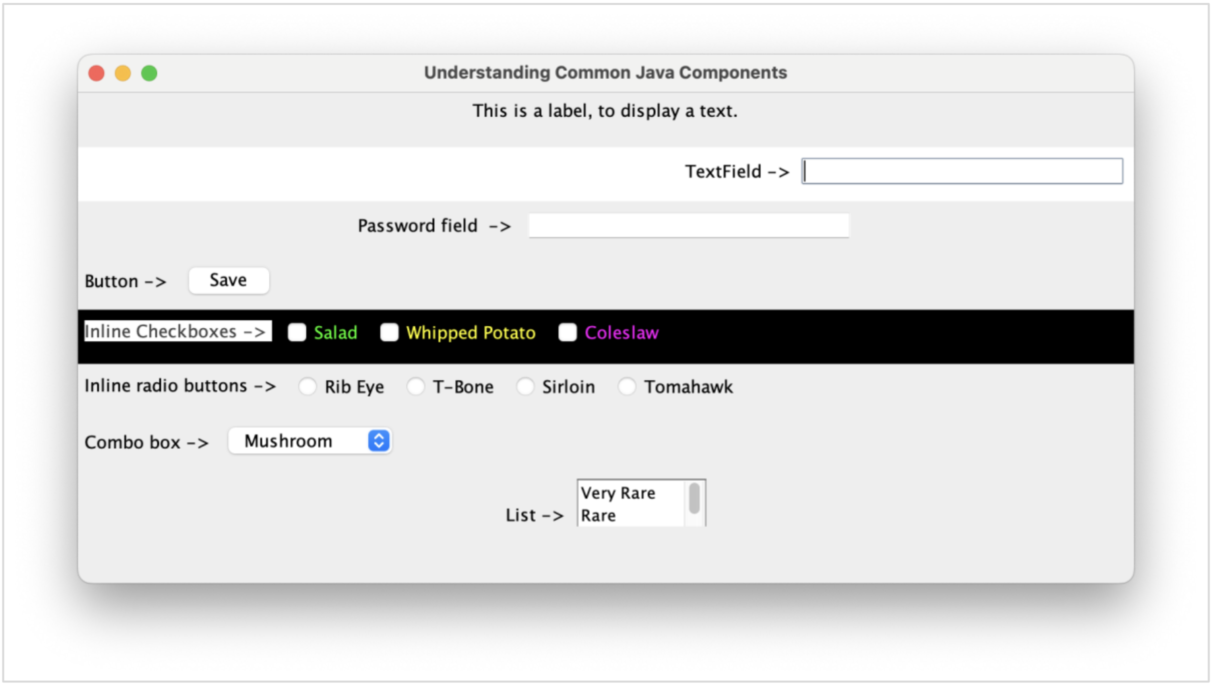

Figure 4: Common GUI components for Java desktop application

Exercise 5: GUI Component Identification

a. Open the **ComponentScreen.java** file.
b. Observe the code and the output from the execution.
c. Identify the object and its constructor for each of the GUI components loaded on the screen from MainGUIApp.java execution. Record the answer in Table 2.

Table 2: GUI component observation results

GUI Component	Object	Constructor

Exercise 6: Overriding GUI Component Color

a. Describe how should a programmer change the color for the GUI components.
b. Based on your description, change the default color for one of the GUI components to the color of your choice.

Exercise 7: Relocate Initial Screen Display

a. Amend the class to display the screen in the center of the screen.
b. Highlight the code to show your solution.

Exercise 8: Layout Manager Identification

a. Identify the classes that contains the term Layout from the code.
b. These are known as layout manager classes. A layout manager class arrange the GUI components layout on a panel or frame. List the layout object and classes in the in Table 3.

Table 3: Layout manager class observation results

Object	**Package**	**Constructor**

Exercise 9: Observing Layout Manager Behavior

The purpose of layout manager is to hold a group of GUI components and arrange it according to layout policy design for a particular manager class.

a. Observe the arrangement of GUI components in each panel.
b. Record your observation in Table 4.

Table 4: Panel arrangement observation results

Panel	**Observation**
Label panel	
Text field panel	
Password field panel	
Button panel	

Check boxes panel	
Radio buttons panel	
Combo box panel	
List panel	

Exercise 10: Concluding Layout Behavior

Draw some conclusions based on the observation results in Exercise 9. Record your observation in ulearn.

Section 3: Distance Converter Application Development

This section will develop an application called Distance Converter Application. The application is a single-screen application that receives in a distance input in kilometer and imperial conversion unit from the user. The initial look of the application is shown in Figure 5.

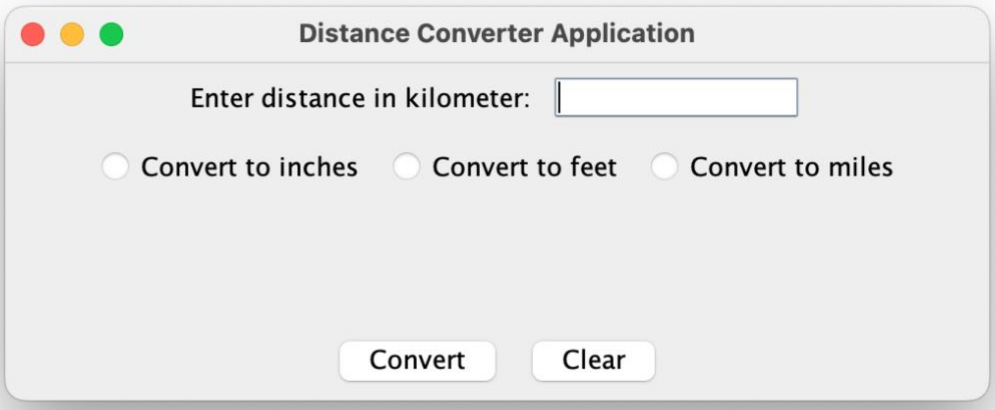

Figure 5: Distance Convertor Application window

Exercise 11: Java Distance Converter Window Definition

1. Create a new window named **Distance Converter Application** using the knowledge and skill comprehended from the previous exercises.
2. The window should be launched from a class named `ConverterApp`.

Exercise 12: User Input Panel Definition

The application consists of three panels. The first panel contains a label and a text field.

1. Define and construct a private object of `JTextField` named `txtDistance` as member of the class defined in Exercise 4. This object will be displayed in the window defined in that class.
2. Create a private method named `getUserInputPanel()`. This method shall return an object of `JPanel`.
3. Provide the implementation of the method using the Java code shown in Figure 6.

```
// A panel to hold GUI component
JPanel userInputPanel = new JPanel();
```

```
// Define label and add to panel
JLabel lblDistance = new JLabel("Enter distance in kilometer:");
userInputPanel.add(lblDistance);

// Set text field initial column size and add  to panel
txtDistance.setColumns(10);
userInputPanel.add(txtDistance);

return userInputPanel;
```

Figure 6: Java code for the method getUserInputPanel ( ) implementation

4. Provide an appropriate comment for method.
5. Add the code snippets shown in Figure 7 before the window is visible.

```
// Add panel to the NORTH region
JPanel userInputPanel = getUserInputPanel();
super.add(userInputPanel);
```

Figure 7: Code snippet to add user input panel to the window

6. Import the necessary classes.
7. Fix any errors.
8. Save the class.
9. Execute the window from class `ConverterApp` defined in the previous exercise. The output shall be similar as shown in Figure 8.

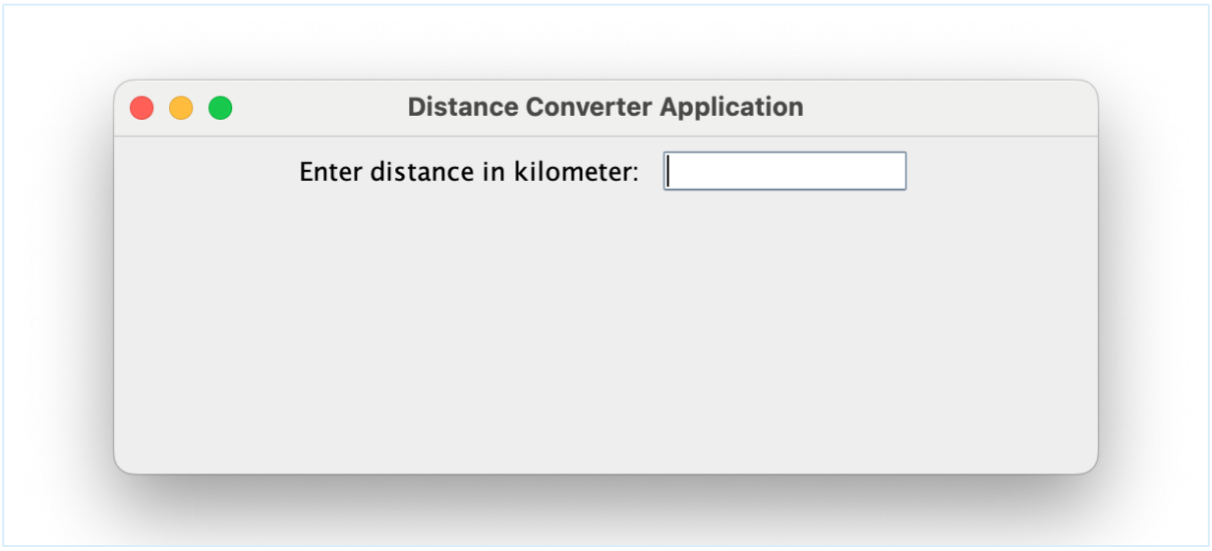

Figure 8: Expected output from Exercise 5

Exercise 13: User Selection Panel Definition

Using the knowledge and skill comprehended in from the previous exercises, create a panel to lay three (3) radio buttons as shown in Figure 5, using `JRadioButtons`. Group the radio buttons using `ButtonGroup` class.

The solution must be neat, readable, easy to read and comply to object-oriented principles.

Exercise 14: Button Panel

Using the knowledge and skill comprehended in from the previous exercises, create a panel to lay two (2) buttons as shown in Figure 5 using `JButtons`.

The solution must be neat, readable, easy to read and comply to object -oriented principles.

Exercise 15: Screen Launcher

Create a class to display the screen created from the previous exercises in this section using the knowledge and skill comprehended in from the previous exercises. The solution must be neat, readable, easy to read and comply to object -oriented principles.

Submit your solutions and screen shot output in ulearn.

End of Document