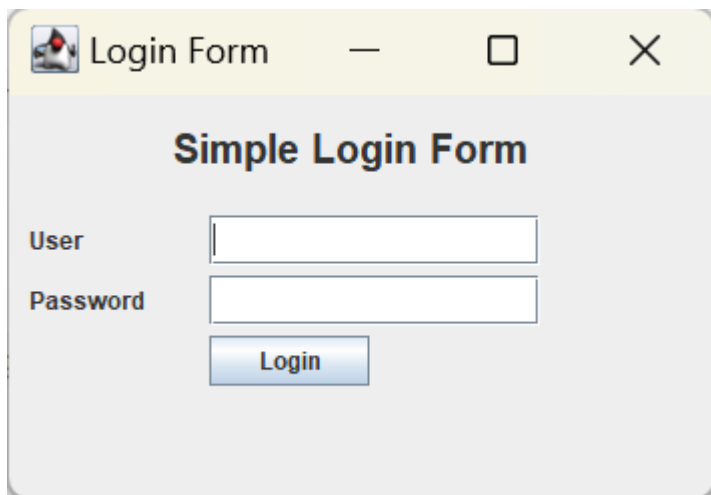


## Programming 2

### Tutorial 13

#### Exercise 1: (Required)

Design the employee management interface as shown in the following image



GUIDELINES:

- ✚ Import Necessary Libraries

```
import javax.swing.*;  
import java.awt.*;  
import java.awt.event.*;
```

- ✚ Declare Class and Components

- The loginGui class is declared, which implements the ActionListener interface to handle button click events.

```

public class loginGui implements ActionListener {
    5 usages
    private static JLabel titleLabel;
    3 usages
    private static JLabel userLabel;
    4 usages
    private static JTextField userText;
    3 usages
    private static JLabel passLabel;
    4 usages
    private static JPasswordField passText;
    4 usages
    private static JButton button;
    5 usages
    private static JLabel success;

```

### Create Main Method

- The main method is the entry point of the application. Inside this method:
- A JFrame named "Login Form" is created.
- A JPanel named panel is created to hold the components.
- The size, default close operation, and panel are set for the frame.

```

public static void main(String[] args) {
    JFrame frame = new JFrame( title: "Login Form");
    JPanel panel = new JPanel();

    frame.setSize( width: 350, height: 230); // Increased height to accommodate the title
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    frame.add(panel);

    panel.setLayout(null);

```

### Create and Configure Title Label

- A JLabel named titleLabel is created to display the title "Simple Login Form". The label is set to:
- Be centered within the panel.
- Use a bold Arial font with a size of 20.

```
// Title label
titleLabel = new JLabel( text: "Simple Login Form");
titleLabel.setBounds( x: 10, y: 10, width: 320, height: 35); // Set position and size
titleLabel.setHorizontalAlignment(JLabel.CENTER); // Center align text
titleLabel.setFont(new Font( name: "Arial", Font.BOLD, size: 20)); // Set font size and style
panel.add(titleLabel);
```

### ✚ Create and Configure User Components

- Components for entering the username are created:
- userLabel: A label displaying "User".
- userText: A text field for entering the username.

```
userLabel = new JLabel( text: "User");
userLabel.setBounds( x: 10, y: 60, width: 80, height: 25);
panel.add(userLabel);

userText = new JTextField( columns: 20);
userText.setBounds( x: 100, y: 60, width: 165, height: 25);
panel.add(userText);
```

### ✚ Create and Configure Password Components

- Components for entering the password are created:
- passLabel: A label displaying "Password".
- passText: A password field for entering the password.

```
passLabel = new JLabel( text: "Password");
passLabel.setBounds( x: 10, y: 90, width: 80, height: 25);
panel.add(passLabel);

passText = new JPasswordField();
passText.setBounds( x: 100, y: 90, width: 165, height: 25);
panel.add(passText);
```

### ✚ Create and Configure Login Button

- A JButton named button is created with the text "Login". An action listener is added to the button to handle button click events.

```
button = new JButton( text: "Login");
button.setBounds( x: 100, y: 120, width: 80, height: 25);
button.addActionListener(new loginGui());
panel.add(button);
```

#### Create and Configure Success Label ( optional)

- A JLabel named success is created to display the login status message.

```
success = new JLabel( text: "");
success.setBounds( x: 10, y: 150, width: 320, height: 25);
panel.add(success);
```

#### Implement ActionListener ( Optional)

- The actionPerformed method is implemented to handle the button click event. It retrieves the entered username and password, checks if they match the expected values, and updates the success label accordingly.

```
@Override
public void actionPerformed(ActionEvent e) {
    String user = userText.getText();
    String password = String.valueOf(passText.getPassword());

    if (user.equals("Karl") && password.equals("123")) {
        success.setText("Login Successful!");
    } else {
        success.setText("Login Failed. Please try again.");
    }
}
```

#### Run the Application

- Finally, the main method ends by making the frame visible, allowing you to interact with the Login Form.

```
frame.setVisible(true);
```