**Title**

**Create a GUI to show “Welcome to Java World” pop up message when GUI is launched.**

**Objective  
To learn how to create a GUI that displays Welcome to Java World.**

**Program code:**

import javax.swing.\*;

class Main

{

public static void main(String[] args)

{

JFrame frame = new JFrame("Lab1");

frame.setSize(500,500);

frame.setVisible(true);  
 frame.setLayout(null);

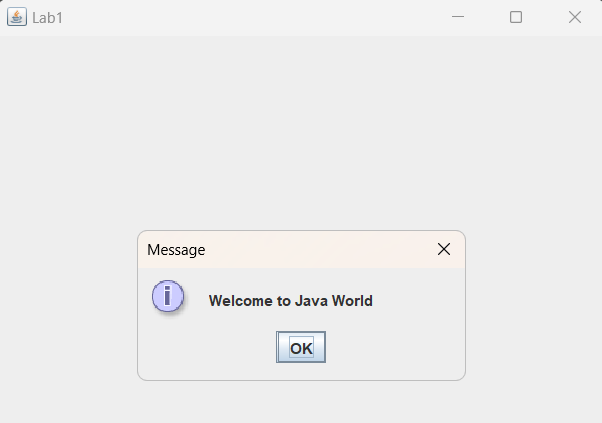
frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

JOptionPane.showMessageDialog(frame,"Welcome to Java World");

}

}

**OUTPUT:**



**Discussion**

**The practice for making a GUI that displays the message as passed in the program was understood and performed successfully.**

**Conclusion:**

**By completing this program, I learnt to create a GUI and display desired content.**

**Title**

**Create a GUI having two textbox that adds both values when button is clicked.**

**Objective  
To learn how to create a GUI with two textbox and a button which performs addition when button is clicked.**

**Program Code:**import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Main {

public static void main(String[] args) {

JFrame f1 = new JFrame("Program2");

f1.setSize(300, 250);

f1.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

f1.setLayout(null);

f1.setVisible(true);

JLabel label1 = new JLabel("Value 1:");

label1.setBounds(30, 30, 80, 30);

f1.add(label1);

JTextField textField1 = new JTextField();

textField1.setBounds(110, 30, 100, 30);

f1.add(textField1);

JLabel label2 = new JLabel("Value 2:");

label2.setBounds(30, 70, 80, 30);

f1.add(label2);

JTextField textField2 = new JTextField();

textField2.setBounds(110, 70, 100, 30);

f1.add(textField2);

JButton button1 = new JButton("Add");

button1.setBounds(110, 110, 100, 30);

f1.add(button1);

JLabel label3 = new JLabel("Result:");

label3.setBounds(30, 150, 200, 30);

f1.add(label3);

button1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try {

int num1 = Integer.parseInt(textField1.getText());

int num2 = Integer.parseInt(textField2.getText());

int sum = num1 + num2;

resultLabel.setText("Result: " + sum);

} catch (NumberFormatException ex) {

resultLabel.setText("Please enter valid integers.");

}

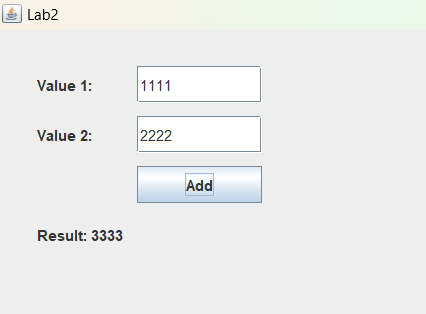
}

});

}

}

**OUTPUT:**

****

**Discussion**

**The practice for making a GUI consisting two textbox which took inputs & performed addition on button click in the program was understood and performed successfully.**

**Conclusion:**

**By completing this program, I learnt to create a GUI with textbox and action listener and performing desired actions on click.**

**Title**

**Create a GUI having two buttons and swap the text when button is clicked.**

**Objective  
To learn how to create a GUI with two buttons and button click swaps the value among the buttons on click.**

**Program Code:**

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Main {

public static void main(String[] args) {

JFrame f1 = new JFrame("Lab3");

f1.setSize(300, 200);

f1.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

f1.setLayout(null);

f1.setVisible(true);

JButton button1 = new JButton("Roronoa");

button1.setBounds(50, 50, 100, 30);

f1.add(button1);

JButton button2 = new JButton("Zoro");

button2.setBounds(160, 50, 100, 30);

f1.add(button2);

button1.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String temp = button1.getText();

button1.setText(button2.getText());

button2.setText(temp);

}

});

button2.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String temp = button1.getText();

button1.setText(button2.getText());

button2.setText(temp);

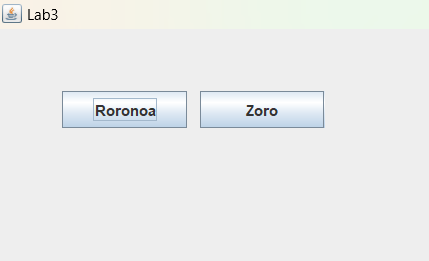
}

});

}

}

**OUTPUT:**


**Discussion**

**The practice for making a GUI having two buttons which swapped each other values onclick was understood and performed successfully.**

**Conclusion:**

**By completing this program, I learnt to create a GUI consisting buttons which swapped values onclick.**

**Title**  
**Create a GUI of a calculator having four button(+,-,\*,/) and perform related arithmetic operation on values when button is clicked.**

**Objective  
To learn how to create a GUI of a calculator having all arithmetic operations performed on values when button is clicked.**

**Program Code:**

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Calculator extends JFrame implements ActionListener {

private JTextField value1Field, value2Field;

private JLabel resultLabel;

private JButton addButton, subButton, mulButton, divButton;

public Calculator() {

createUI();

}

private void createUI() {

setTitle("Lab4");

setSize(400, 400);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

setLayout(null);

Font labelFont = new Font("Arial", Font.PLAIN, 14);

JLabel value1Label = new JLabel("Value 1:");

value1Label.setBounds(50, 30, 100, 30);

value1Label.setFont(labelFont);

value1Field = new JTextField();

value1Field.setBounds(150, 30, 200, 30);

value1Field.setFont(labelFont);

JLabel value2Label = new JLabel("Value 2:");

value2Label.setBounds(50, 70, 100, 30);

value2Label.setFont(labelFont);

value2Field = new JTextField();

value2Field.setBounds(150, 70, 200, 30);

value2Field.setFont(labelFont);

resultLabel = new JLabel("Result: ");

resultLabel.setBounds(50, 110, 300, 30);

resultLabel.setFont(labelFont);

Font buttonFont = new Font("Arial", Font.BOLD, 16);

addButton = new JButton("+");

subButton = new JButton("-");

mulButton = new JButton("\*");

divButton = new JButton("/");

addButton.setBounds(50, 160, 60, 50);

subButton.setBounds(130, 160, 60, 50);

mulButton.setBounds(210, 160, 60, 50);

divButton.setBounds(290, 160, 60, 50);

addButton.setFont(buttonFont);

subButton.setFont(buttonFont);

mulButton.setFont(buttonFont);

divButton.setFont(buttonFont);

add(value1Label);

add(value1Field);

add(value2Label);

add(value2Field);

add(resultLabel);

add(addButton);

add(subButton);

add(mulButton);

add(divButton);

addButton.addActionListener(this);

subButton.addActionListener(this);

mulButton.addActionListener(this);

divButton.addActionListener(this);

}

@Override

public void actionPerformed(ActionEvent e) {

try {

int firstNumber = Integer.parseInt(value1Field.getText());

int secondNumber = Integer.parseInt(value2Field.getText());

int result = 0;

switch (e.getActionCommand()) {

case "+":

result = firstNumber + secondNumber;

break;

case "-":

result = firstNumber - secondNumber;

break;

case "\*":

result = firstNumber \* secondNumber;

break;

case "/":

if (secondNumber != 0) {

result = firstNumber / secondNumber;

} else {

JOptionPane.showMessageDialog(null, "Cannot divide by zero");

return;

}

break;

}

resultLabel.setText("Result: " + result);

} catch (NumberFormatException ex) {

JOptionPane.showMessageDialog(null, "Please enter valid numbers");

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

new Calculator().setVisible(true);

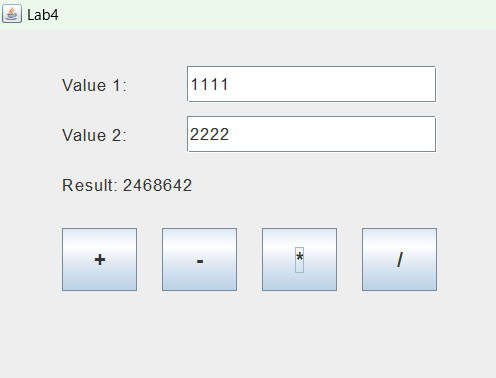
}

});

}

}

**OUTPUT:**



**Discussion  
The practice for making a GUI of a calculator which performed arithmetic operations on values when button clicked was understood and performed successfully.**

**Conclusion:**

**By completing this program, I learnt to create a GUI of a calculator and performed desired arithmetic operations.**

**Title**

**Create a simple calculation that takes input in a box and display results in same box.**

**Objective  
To learn how to create a GUI of a calculator having all input values in a box which is passed on click of input value buttons and arithmetic operations is performed on the button clicks and result displayed.**

**Program Code:**

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Main {

JFrame f1;

JTextField display;

JButton b1, b2, b3, b4, b5, b6, b7, b8, b9, plus, minus, mul, div, zero, clear, res;

double num1 = 0, num2 = 0, result = 0;

char operator;

public Main() {

f1 = new JFrame("Lab5");

f1.setSize(350, 400);

f1.setLayout(null);

f1.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

display = new JTextField();

display.setBounds(50, 20, 250, 50);

display.setEditable(false);

b1 = createButton("1", 50, 85);

b2 = createButton("2", 115, 85);

b3 = createButton("3", 180, 85);

b4 = createButton("4", 50, 130);

b5 = createButton("5", 115, 130);

b6 = createButton("6", 180, 130);

b7 = createButton("7", 50, 175);

b8 = createButton("8", 115, 175);

b9 = createButton("9", 180, 175);

zero = createButton("0", 115, 225);

plus = createButton("+", 245, 85);

minus = createButton("-", 245, 130);

mul = createButton("\*", 245, 175);

div = createButton("/", 245, 225);

clear = createButton("C", 50, 225);

res = createButton("=", 180, 225);

f1.add(display);

f1.setVisible(true);

}

private JButton createButton(String text, int x, int y) {

JButton button = new JButton(text);

button.setBounds(x, y, 50, 40);

button.addActionListener(new ButtonClickListener());

f1.add(button);

return button;

}

private class ButtonClickListener implements ActionListener {

public void actionPerformed(ActionEvent e) {

String command = e.getActionCommand();

if ((command.charAt(0) >= '0' && command.charAt(0) <= '9') || command.charAt(0) == '.') {

display.setText(display.getText() + command);

} else if (command.charAt(0) == 'C') {

display.setText("");

num1 = num2 = result = 0;

operator = '\0';

} else if (command.charAt(0) == '=') {

num2 = Double.parseDouble(display.getText());

switch (operator) {

case '+':

result = num1 + num2;

break;

case '-':

result = num1 - num2;

break;

case '\*':

result = num1 \* num2;

break;

case '/':

result = num1 / num2;

break;

}

display.setText(String.valueOf(result));

} else {

if (!display.getText().isEmpty()) {

num1 = Double.parseDouble(display.getText());

operator = command.charAt(0);

display.setText("");

}

}

}

}

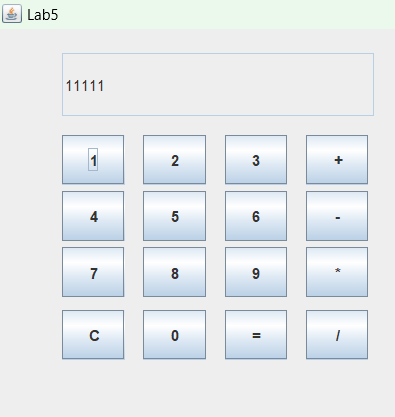
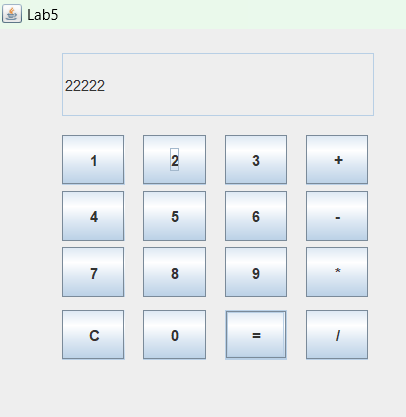
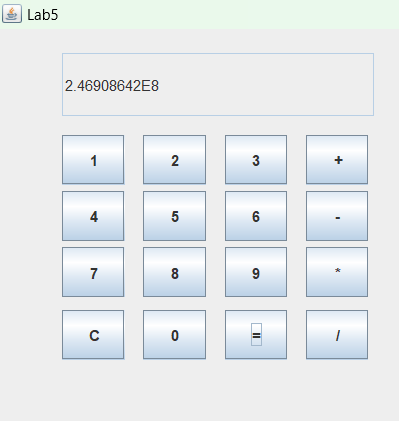
public static void main(String[] args) {

new Main();

}

}

**OUTPUT:**

**Discussion  
The practice for making a GUI of a calculator which performed arithmetic operations on values which are passed through input value buttons button clicked was understood and performed successfully.**

**Conclusion:**

**By completing this program, I learnt to create a GUI of a calculator which had all input values as button inputs and performed desired arithmetic operations.**

**Title  
Create a gallery having some thumbnails with action listener and show full screen image when clicked.**

**Objective  
To learn how to create a gallery of desired images thumbnails and on click the desired images to be displayed on full screen.**

**Program Code:**

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Main {

public static void main(String[] args) {

JFrame f1 = new JFrame("Lab6");

f1.setLayout(null);

f1.setSize(800, 800);

f1.setVisible(true);

f1.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

ImageIcon ic1 = new ImageIcon("C:\\Users\\HP\\Downloads\\ace.png");

ImageIcon ic2 = new ImageIcon("C:\\Users\\HP\\Downloads\\luffy.png");

ImageIcon ic3 = new ImageIcon("C:\\Users\\HP\\Downloads\\sanji.png");

ImageIcon ic4 = new ImageIcon("C:\\Users\\HP\\Downloads\\zoro.png");

JButton b1 = new JButton(ic1);

JButton b2 = new JButton(ic2);

JButton b3 = new JButton(ic3);

JButton b4 = new JButton(ic4);

b1.setBounds(0, 0, 400, 400);

b2.setBounds(400, 0, 400, 400);

b3.setBounds(0, 400, 400, 400);

b4.setBounds(400, 400, 400, 400);

b1.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

displayZoomedImage(ic1);

}

});

b2.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

displayZoomedImage(ic2);

}

});

b3.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

displayZoomedImage(ic3);

}

});

b4.addActionListener(new ActionListener() {

public void actionPerformed(ActionEvent e) {

displayZoomedImage(ic4);

}

});

f1.add(b1);

f1.add(b2);

f1.add(b3);

f1.add(b4);

}

private static void displayZoomedImage(ImageIcon icon) {

JFrame f2 = new JFrame("Zoomed Image");

f2.setSize(800, 800);

JLabel l2 = new JLabel(icon);

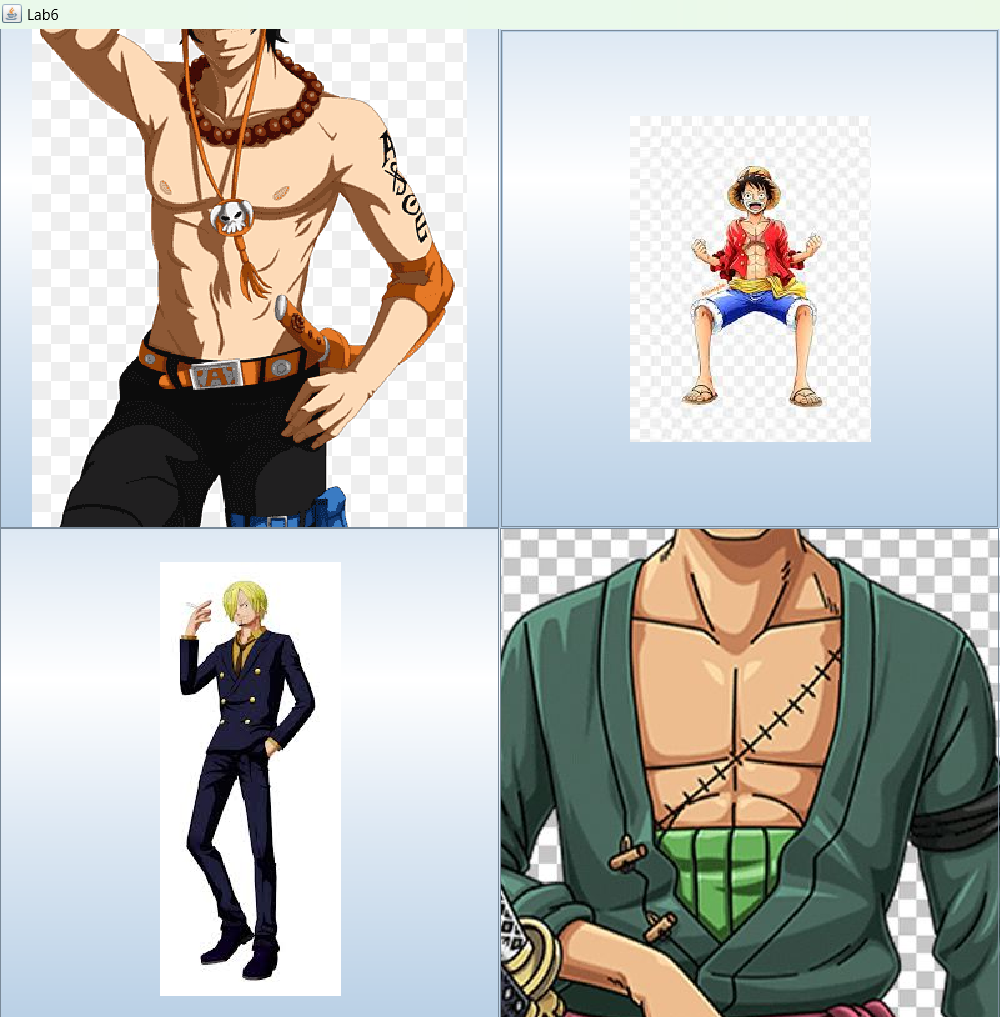
f2.add(l2);

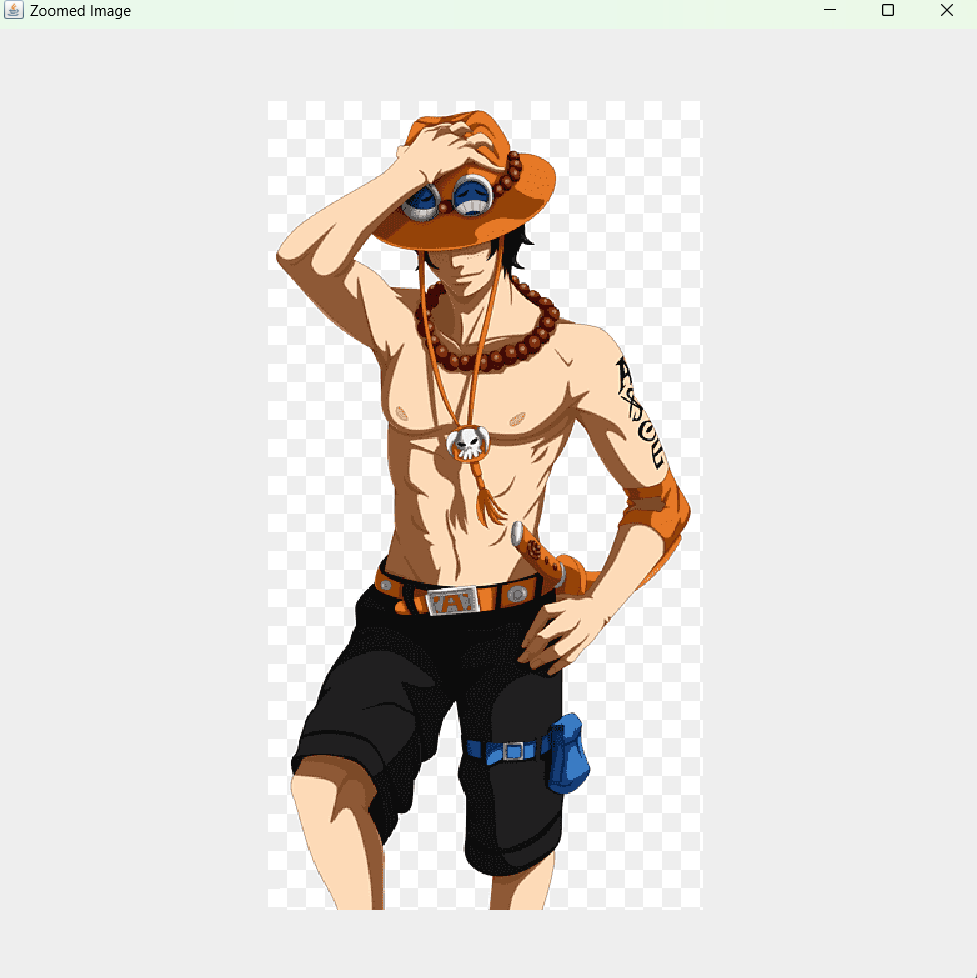
f2.setVisible(true);

}

}

**OUTPUT:**

****



**Discussion  
The practice for making a GUI of a gallery of desired images thumbnails and on click the desired images to be displayed on full screen with the help of action listener was understood and performed successfully.**

**Conclusion:**

**By completing this program, I learnt to create a gallery of images and display desire image on full screen via the help of action listener.**

**Title**

**Create a login form with database verification.**

**Objective  
To learn to verify an user login from the existing records in the database.**

**Program Code:**import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

public class Databaselab1 extends JFrame {

private JTextField usernameField;

private JPasswordField passwordField;

private JButton loginButton;

private JPanel mainPanel;

public Databaselab1() {

setTitle("Database Lab1 ");

setSize(350, 250);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

mainPanel = new JPanel();

mainPanel.setLayout(null);

JLabel userLabel = new JLabel("Username:");

userLabel.setBounds(30, 30, 80, 25);

mainPanel.add(userLabel);

usernameField = new JTextField(20);

usernameField.setBounds(120, 30, 185, 25);

mainPanel.add(usernameField);

JLabel passwordLabel = new JLabel("Password:");

passwordLabel.setBounds(30, 70, 80, 25);

mainPanel.add(passwordLabel);

passwordField = new JPasswordField(20);

passwordField.setBounds(120, 70, 185, 25);

mainPanel.add(passwordField);

loginButton = new JButton("Login");

loginButton.setBounds(30, 110, 275, 30);

mainPanel.add(loginButton);

add(mainPanel);

loginButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try {

Connection connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/datalist", "root", "");

String query = "SELECT \* FROM datalist WHERE Usernames=? AND Passwords=?";

PreparedStatement preparedStatement = connection.prepareStatement(query);

preparedStatement.setString(1, usernameField.getText());

preparedStatement.setString(2, new String(passwordField.getPassword()));

ResultSet resultSet = preparedStatement.executeQuery();

if (resultSet.next()) {

JOptionPane.showMessageDialog(null, "Login successful!");

} else {

JOptionPane.showMessageDialog(null, "Invalid username or password.");

}

connection.close();

} catch (Exception ex) {

ex.printStackTrace();

}

}

});

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

new Databaselab1().setVisible(true);

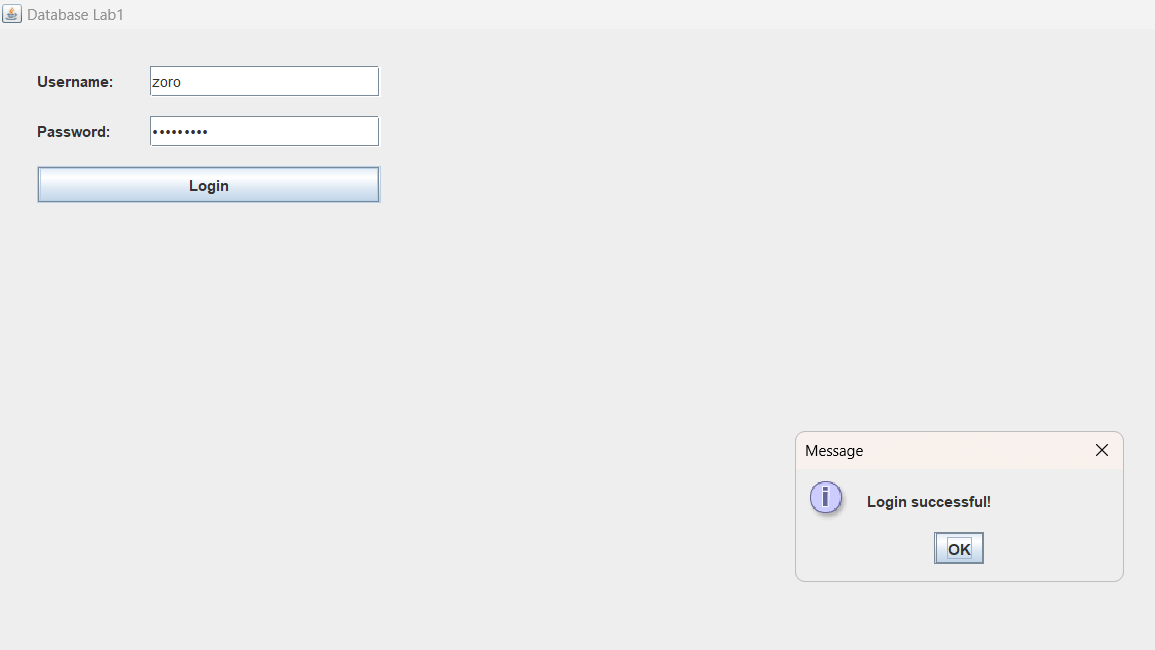
}

});

}

}

**OUTPUT:**



**Discussion  
The practice for validating an user login from the existing records in the database which was accessed through the sqlconnector library imported in the program was understood and performed successfully.**

**Conclusion:**

**By completing this program, I learnt to validate an user login with the existing records in the database.**

**Title**

**Create a registration form containing text fields, passwords, radio button, checkbox, label, button with confirm password field which should be validated.**

**Objective  
To learn to confirm the two user input passwords and register the user only if the two passwords match each other in the form along with other details.**

**Program Code:**

import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class Databaselab2 extends JFrame {

private JTextField usernameField;

private JPasswordField passwordField;

private JPasswordField confirmPasswordField;

private JRadioButton maleRadioButton;

private JRadioButton femaleRadioButton;

private JCheckBox agreeCheckBox;

private JButton registerButton;

private JPanel mainPanel;

public Databaselab2() {

setTitle("Database Lab2");

setSize(400, 350);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

mainPanel = new JPanel();

mainPanel.setLayout(null);

JLabel userLabel = new JLabel("Username:");

userLabel.setBounds(30, 30, 80, 25);

mainPanel.add(userLabel);

usernameField = new JTextField(20);

usernameField.setBounds(150, 30, 200, 25);

mainPanel.add(usernameField);

JLabel passwordLabel = new JLabel("Password:");

passwordLabel.setBounds(30, 70, 80, 25);

mainPanel.add(passwordLabel);

passwordField = new JPasswordField(20);

passwordField.setBounds(150, 70, 200, 25);

mainPanel.add(passwordField);

JLabel confirmPasswordLabel = new JLabel("Confirm Password:");

confirmPasswordLabel.setBounds(30, 110, 120, 25);

mainPanel.add(confirmPasswordLabel);

confirmPasswordField = new JPasswordField(20);

confirmPasswordField.setBounds(150, 110, 200, 25);

mainPanel.add(confirmPasswordField);

JLabel genderLabel = new JLabel("Gender:");

genderLabel.setBounds(30, 150, 80, 25);

mainPanel.add(genderLabel);

maleRadioButton = new JRadioButton("Male");

maleRadioButton.setBounds(150, 150, 80, 25);

mainPanel.add(maleRadioButton);

femaleRadioButton = new JRadioButton("Female");

femaleRadioButton.setBounds(230, 150, 80, 25);

mainPanel.add(femaleRadioButton);

ButtonGroup genderGroup = new ButtonGroup();

genderGroup.add(maleRadioButton);

genderGroup.add(femaleRadioButton);

agreeCheckBox = new JCheckBox("I agree to the terms and conditions.");

agreeCheckBox.setBounds(30, 190, 320, 25);

mainPanel.add(agreeCheckBox);

registerButton = new JButton("Register");

registerButton.setBounds(150, 230, 100, 30);

mainPanel.add(registerButton);

add(mainPanel);

registerButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String password = new String(passwordField.getPassword());

String confirmPassword = new String(confirmPasswordField.getPassword());

if (!password.equals(confirmPassword)) {

JOptionPane.showMessageDialog(null, "Passwords do not match!");

} else if (!agreeCheckBox.isSelected()) {

JOptionPane.showMessageDialog(null, "You must agree to the terms and conditions.");

} else {

JOptionPane.showMessageDialog(null, "Registration successful!");

}

}

});

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

new Databaselab2().setVisible(true);

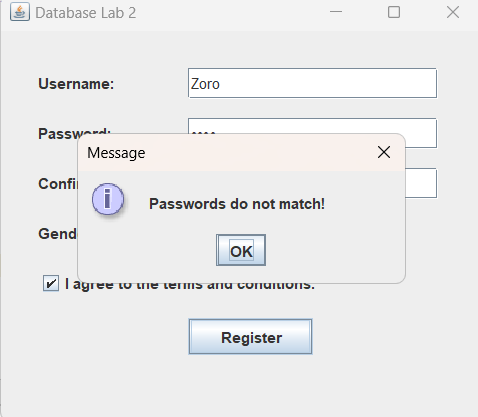
}

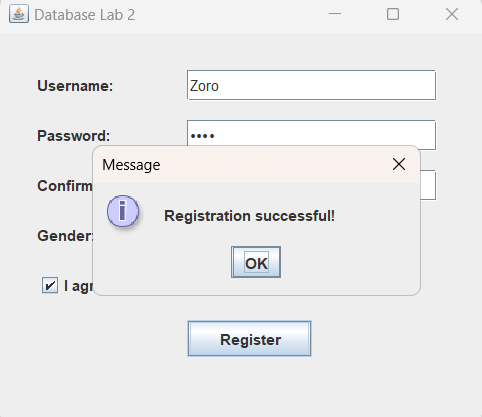
});

}

}

**OUTPUT:**





**Discussion  
The practice for registering an user only and only if the two passwords input in the password field and confirm password field match each other and if they don’t match return the error message was understood and performed successfully.**

**Conclusion:**

**By completing this program, I learnt to validate the user password inputs and confirm passwords input comparing them to each other and only registering in the case of successful password inputs.**

**Title  
 Create a GUI that displays record from database.**

**Objective  
To learn to access the database and view the database existing records.**

**Program Code:**import javax.swing.\*;

import javax.swing.table.DefaultTableModel;

import java.awt.\*;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

public class Databaselab3 extends JFrame {

private JTable table;

private DefaultTableModel tableModel;

public Databaselab3() {

setTitle("Program 9");

setSize(600, 400);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

String[] columnNames = {"Username", "Password"};

tableModel = new DefaultTableModel(columnNames, 0);

table = new JTable(tableModel);

JScrollPane scrollPane = new JScrollPane(table);

scrollPane.setBounds(20, 20, 550, 300);

JPanel mainPanel = new JPanel();

mainPanel.setLayout(null);

mainPanel.add(scrollPane);

add(mainPanel);

loadDataFromDatabase();

}

private void loadDataFromDatabase() {

String url = "jdbc:mysql://localhost:3306/datalist?zeroDateTimeBehavior=convertToNull";

String user = "root";

String password = "";

try {

Connection connection = DriverManager.getConnection(url, user, password);

Statement statement = connection.createStatement();

String query = "SELECT \* FROM datalist";

ResultSet resultSet = statement.executeQuery(query);

while (resultSet.next()) {

String username = resultSet.getString("Usernames");

String passwords = resultSet.getString("Passwords");

tableModel.addRow(new Object[]{username, passwords});

}

connection.close();

} catch (Exception ex) {

ex.printStackTrace();

}

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

new Databaselab3().setVisible(true);

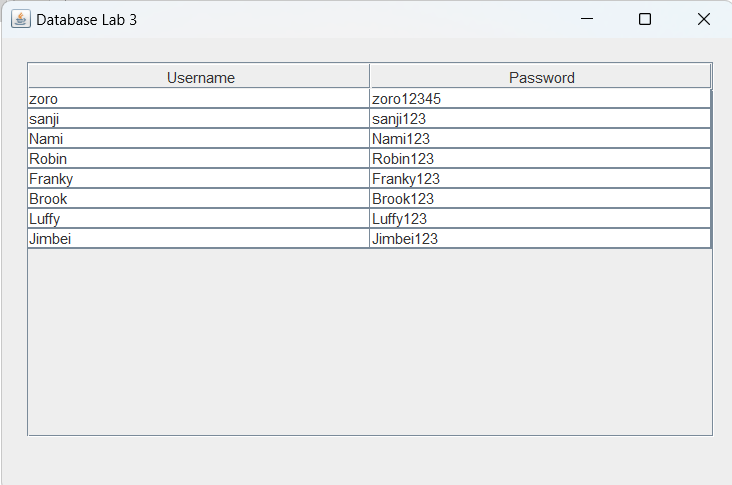
}

});

}

}

**OUTPUT:**



**Discussion  
The practice for accessing the database with the help of sqlconnector imported in the library and making the view for the existing database records in a tabular form was understood and performed successfully.**

**Conclusion:  
By completing this program, I learnt to retrieve the data existing in the database with the help of required libraries and present it in a tabular form.**

**Title  
Create a GUI that will receive password of an existing user and change it in database when button is clicked.**

**Objective  
To learn to retrieve the password of an existing user in the database and change the existing password to a new password in the database through the input textfield.**

**Program Code:**import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

public class Databaselab4 extends JFrame {

private JTextField usernameField;

private JPasswordField newPasswordField;

private JButton changePasswordButton;

private JPanel mainPanel;

public Databaselab4() {

setTitle("Database Lab 4");

setSize(400, 250);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

mainPanel = new JPanel();

mainPanel.setLayout(null);

JLabel usernameLabel = new JLabel("Username:");

usernameLabel.setBounds(30, 30, 100, 25);

mainPanel.add(usernameLabel);

usernameField = new JTextField(20);

usernameField.setBounds(150, 30, 200, 25);

mainPanel.add(usernameField);

JLabel newPasswordLabel = new JLabel("New Password:");

newPasswordLabel.setBounds(30, 70, 100, 25);

mainPanel.add(newPasswordLabel);

newPasswordField = new JPasswordField(20);

newPasswordField.setBounds(150, 70, 200, 25);

mainPanel.add(newPasswordField);

changePasswordButton = new JButton("Change Password");

changePasswordButton.setBounds(100, 120, 200, 30);

mainPanel.add(changePasswordButton);

add(mainPanel);

changePasswordButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String username = usernameField.getText();

String newPassword = new String(newPasswordField.getPassword());

String url = "jdbc:mysql://localhost:3306/datalist?zeroDateTimeBehavior=convertToNull";

String user = "root";

String password = "";

try {

Connection connection = DriverManager.getConnection(url, user, password);

String fetchQuery = "SELECT Passwords FROM datalist WHERE Usernames=?";

PreparedStatement fetchStmt = connection.prepareStatement(fetchQuery);

fetchStmt.setString(1, username);

ResultSet resultSet = fetchStmt.executeQuery();

if (resultSet.next()) {

String oldPassword = resultSet.getString("Passwords");

String updateQuery = "UPDATE datalist SET Passwords=? WHERE Usernames=?";

PreparedStatement updateStmt = connection.prepareStatement(updateQuery);

updateStmt.setString(1, newPassword);

updateStmt.setString(2, username);

int rowsUpdated = updateStmt.executeUpdate();

if (rowsUpdated > 0) {

JOptionPane.showMessageDialog(null, "Password for user: " + username + " has been changed from \"" + oldPassword + "\" to \"" + newPassword + "\".");

} else {

JOptionPane.showMessageDialog(null, "Failed to update password. User not found.");

}

} else {

JOptionPane.showMessageDialog(null, "User not found.");

}

connection.close();

} catch (Exception ex) {

ex.printStackTrace();

}

}

});

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

new Databaselab4().setVisible(true);

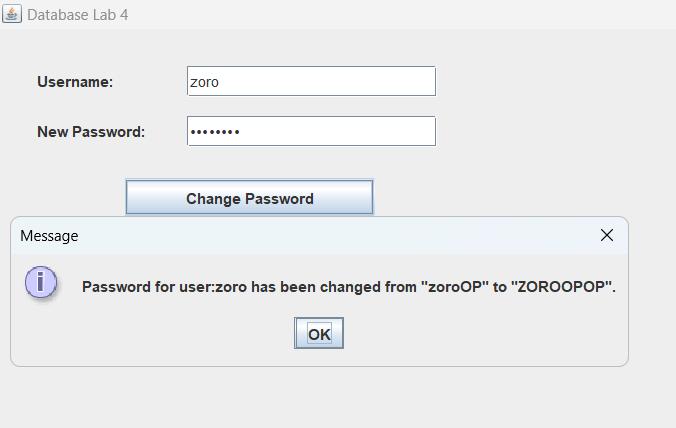
}

});

}

}

**OUTPUT:**



**Discussion  
The practice for accessing the database with the help of sqlconnector imported in the library retrieving the password of existing user and changing it to a new password was understood and performed successfully.**

**Conclusion:  
By completing this program, I learnt to retrieve the password of an existing user in the database and make desired changes to it.**

**Title  
 Create a GUI to ask username and delete that user from the database.**

**Objective  
 To delete an existing user in the database with the help of username.**

**Program Code:**import javax.swing.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

public class Databaselab5 extends JFrame {

private JTextField usernameField;

private JButton deleteButton;

private JPanel mainPanel;

public Databaselab5() {

setTitle("Databaselab 5");

setSize(400, 200);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

mainPanel = new JPanel();

mainPanel.setLayout(null);

JLabel usernameLabel = new JLabel("Username:");

usernameLabel.setBounds(30, 30, 100, 25);

mainPanel.add(usernameLabel);

usernameField = new JTextField(20);

usernameField.setBounds(150, 30, 200, 25);

mainPanel.add(usernameField);

deleteButton = new JButton("Delete User");

deleteButton.setBounds(100, 70, 200, 30);

mainPanel.add(deleteButton);

add(mainPanel);

deleteButton.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

String username = usernameField.getText();

String url = "jdbc:mysql://localhost:3306/datalist?zeroDateTimeBehavior=convertToNull";

String user = "root";

String password = "";

try {

Connection connection = DriverManager.getConnection(url, user, password);

String deleteQuery = "DELETE FROM datalist WHERE Usernames=?";

PreparedStatement deleteStmt = connection.prepareStatement(deleteQuery);

deleteStmt.setString(1, username);

int rowsDeleted = deleteStmt.executeUpdate();

if (rowsDeleted > 0) {

JOptionPane.showMessageDialog(null, "User " + username + " has been deleted.");

} else {

JOptionPane.showMessageDialog(null, "User not found.");

}

connection.close();

} catch (Exception ex) {

ex.printStackTrace();

JOptionPane.showMessageDialog(null, "Error occurred while deleting user.");

}

}

});

}

public static void main(String[] args) {

SwingUtilities.invokeLater(new Runnable() {

@Override

public void run() {

new Databaselab5().setVisible(true);

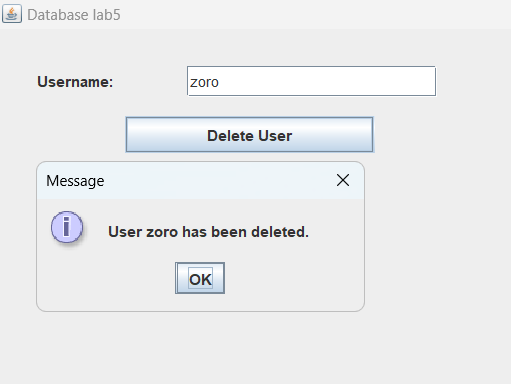
}

});

}

}

**OUTPUT:**



**Discussion  
The practice for accessing the database with the help of sqlconnector imported in the library and delete the existing user from the database with the help of username to that corresponding user was understood and performed successfully.**

**Conclusion:  
By completing this program, I learnt to delete an existing user in the database with the help of username corresponding to that user.**