ASSIGNMENT-8

Date:10/9/2024

Reg No:2022503045

1. Create an interactive Quiz form using radio buttons, check box, text box, text area. Write a method to find the total score of the user. Display the right answer to the user at the end.

CODE:

import javax.swing.\*;

import javax.swing.border.Border;

import java.awt.\*;

import java.awt.event.\*;

public class Quiz\_3045 extends JFrame implements ActionListener {

private JTextField q1;

private JTextArea q2;

private JRadioButton[] q3;

private JCheckBox[] q4;

private JComboBox<String> q5;

private JButton submit;

private int score=0;

public Quiz\_3045(){

q1=new JTextField(20);

q2=new JTextArea(5,10);

q3=new JRadioButton[]{

new JRadioButton("int"),

new JRadioButton("float"),

new JRadioButton("String"),

new JRadioButton("boolean")

};

q4=new JCheckBox[]{

new JCheckBox("private"),

new JCheckBox("public"),

new JCheckBox("protected"),

new JCheckBox("final")

};

String[] options={"new", "this","static","class"};

q5=new JComboBox<>(options);

submit=new JButton("Submit");

setLayout(new GridBagLayout());

setSize(1000,1000);

setVisible(true);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

GridBagConstraints gbc=new GridBagConstraints();

Border border=BorderFactory.createLineBorder(Color.BLACK,2);

q1.setBorder(border);

q2.setBorder(border);

gbc.insets=new Insets(15, 15, 15, 15);

gbc.gridx=0;

gbc.gridy=0;

gbc.anchor=GridBagConstraints.WEST;

add(new Label(" What keyword is used to inherit a class in Java?"),gbc);

gbc.gridx=1; gbc.gridy=0; gbc.weightx=1; gbc.weighty=0; gbc.fill=GridBagConstraints.HORIZONTAL;

add(q1,gbc);

gbc.gridx=0;

gbc.gridy=1;

gbc.anchor=GridBagConstraints.WEST;

add(new Label("Explain the purpose of the super keyword in Java."),gbc);

gbc.gridx=1; gbc.gridy=1; gbc.weightx=1; gbc.weighty=1; gbc.fill=GridBagConstraints.BOTH;

add(new JScrollPane(q2), gbc);

ButtonGroup bg=new ButtonGroup();

JPanel radioJPanel=new JPanel();

radioJPanel.setLayout(new GridLayout(1,q3.length));

gbc.gridx=0;

gbc.gridy=2;

gbc.anchor=GridBagConstraints.WEST;

add(new Label("Which of the following is not a primitive data type in Java?"),gbc);

gbc.gridx=1;

gbc.gridy=2;

for(JRadioButton op:q3){

bg.add(op);

radioJPanel.add(op);

}

add(radioJPanel,gbc);

gbc.gridx=0;

gbc.gridy=3;

gbc.anchor=GridBagConstraints.WEST;

add(new Label("Which of the following are access modifiers in Java? (Select all that apply)"),gbc);

JPanel checkBoxPanel=new JPanel();

checkBoxPanel.setLayout(new GridLayout(1,q4.length));

for(JCheckBox checkBox:q4){

checkBoxPanel.add(checkBox);

}

gbc.gridx=1;

gbc.gridy=3;

add(checkBoxPanel,gbc);

gbc.gridx=0;

gbc.gridy=4;

gbc.anchor=GridBagConstraints.WEST;

add(new Label("Which of the following is used to create an instance of a class in Java?"),gbc);

gbc.gridx=1;

gbc.gridy=4;

gbc.weightx=1;

add(q5,gbc);

gbc.gridx=0;

gbc.gridy=5;

gbc.gridwidth=2;

gbc.fill=GridBagConstraints.HORIZONTAL;

gbc.anchor=GridBagConstraints.CENTER;

gbc.weightx=0;

gbc.weighty=0;

submit.setBackground(Color.BLACK);

submit.setForeground(Color.WHITE);

add(submit,gbc);

submit.addActionListener(this);

pack();

}

public void actionPerformed(ActionEvent e){

String q1\_ans=q1.getText();

String q2\_ans=q2.getText();

if(q1\_ans.equals("extends")){

score+=1;

}

if(q2\_ans.toLowerCase().contains("parent class's variables, methods, or constructors from the child class")){

score+=2;

}

String q3\_ans="";

for(JRadioButton option:q3){

if(option.isSelected()){

q3\_ans=option.getText();

break;

}

}

if(q3\_ans.equals("String")){

score+=1;

}

int checkScore=0;

int option=1;

for(JCheckBox checkBox:q4){

if(option<=3 && checkBox.isSelected()){

checkScore+=1;

}

else if(option>3 && checkBox.isSelected()){

checkScore=0;

}

}

score+=checkScore;

String q5\_ans=(String)q5.getSelectedItem();

if(q5\_ans.equals("new")){

score+=1;

}

String message=String.format("Your Total Score is "+score+"/8");

JOptionPane.showMessageDialog(Quiz\_3045.this, message, "Final Score", JOptionPane.INFORMATION\_MESSAGE);

String answers=String.format("\n ANSWERS \n 1.extends\n 2.The super keyword is used to refer to the parent class's variables, methods, or constructors from the child class.\n3.String\n4.private,public,protected\n5.new\n");

JOptionPane.showMessageDialog(Quiz\_3045.this, answers, "ANSWERS", JOptionPane.INFORMATION\_MESSAGE);

score=0;

}

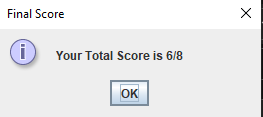
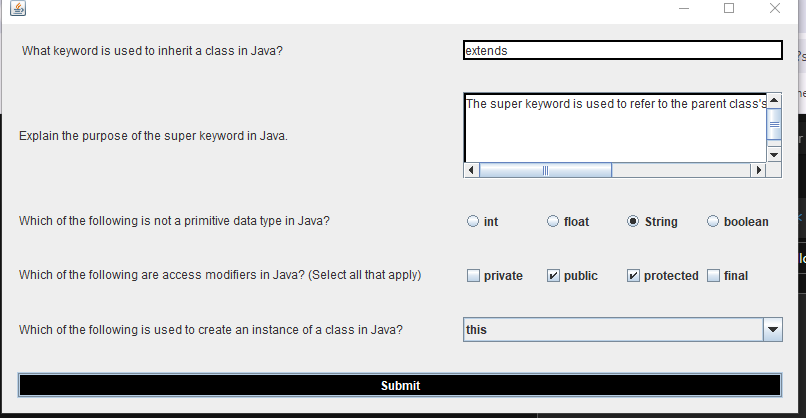
public static void main(String[] args){

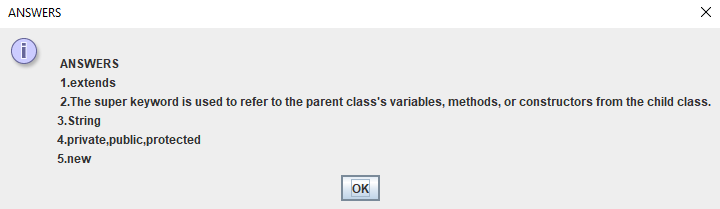
new Quiz\_3045();

}

}

OUTPUT:





2. Create a shopping cart program using Java Swing. The program should allow users to add items to the cart, display the cart's contents (including the items and their prices), and calculate the total amount to be paid. Create a Java Swing application that fulfills these requirements.

CODE:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.util.ArrayList;

public class CartSystem\_3045 extends JFrame {

String[][] items = {

{"Green Women-Maxi dress", "78"},

{"White Women dress", "166"},

{"Men Blue-White shirt", "40"},

{"Summer Women-maxi", "50"},

{"Red Women-Maxi dress", "200"},

{"Men's white shirt", "100"}

};

ArrayList<String[]> cartItems = new ArrayList<>();

String[] paths\_for\_icons = {

"./Items/dress1.jfif",

"./Items/dress2.jfif",

"./Items/dress3.jpg",

"./Items/dress4.jfif",

"./Items/dress5.jfif",

"./Items/dress6.jfif"

};

public CartSystem\_3045() {

setTitle("Shopping Cart");

setSize(800, 600);

setLayout(new BorderLayout());

// Create a panel to hold the items

JPanel itemsPanel = new JPanel(new GridLayout(0, 2, 10, 10));

JScrollPane scrollPane = new JScrollPane(itemsPanel);

add(scrollPane, BorderLayout.CENTER);

// Display each item with its image, name, price, and an "Add to Cart" button

for (int i = 0; i < items.length; i++) {

JPanel itemPanel = new JPanel(new BorderLayout());

// Load and display the image of the item

JLabel itemImageLabel = new JLabel();

ImageIcon icon = new ImageIcon(paths\_for\_icons[i]);

Image scaledImage = icon.getImage().getScaledInstance(100, 100, Image.SCALE\_SMOOTH);

itemImageLabel.setIcon(new ImageIcon(scaledImage));

itemPanel.add(itemImageLabel, BorderLayout.WEST);

// Item name and price

JLabel itemName = new JLabel(items[i][0] + " - $" + items[i][1]);

// Add to Cart button

JButton addToCartButton = new JButton("Add to Cart");

final int index = i;

addToCartButton.addActionListener(e -> {

cartItems.add(items[index]);

JOptionPane.showMessageDialog(null, items[index][0] + " has been added to the cart.");

});

// Add item name and button to the panel

JPanel infoPanel = new JPanel(new BorderLayout());

infoPanel.add(itemName, BorderLayout.CENTER);

infoPanel.add(addToCartButton, BorderLayout.EAST);

itemPanel.add(infoPanel, BorderLayout.CENTER);

itemsPanel.add(itemPanel);

}

// Add a "Display Cart" button and "Checkout" button

JPanel bottomPanel = new JPanel(new FlowLayout());

JButton displayCartButton = new JButton("Display Cart");

displayCartButton.addActionListener(e -> displayCart());

JButton checkoutButton = new JButton("Checkout");

checkoutButton.addActionListener(e -> checkout());

bottomPanel.add(displayCartButton);

bottomPanel.add(checkoutButton);

add(bottomPanel, BorderLayout.SOUTH);

setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

setVisible(true);

}

// Method to display the items in the cart with a remove button

private void displayCart() {

JFrame cartFrame = new JFrame("Cart");

cartFrame.setSize(400, 300);

cartFrame.setLayout(new BorderLayout());

JPanel cartPanel = new JPanel(new GridLayout(0, 1));

JScrollPane cartScrollPane = new JScrollPane(cartPanel);

cartFrame.add(cartScrollPane, BorderLayout.CENTER);

// Display items in the cart with a remove button for each

for (int i = 0; i < cartItems.size(); i++) {

JPanel cartItemPanel = new JPanel(new BorderLayout());

JLabel cartItemLabel = new JLabel(cartItems.get(i)[0] + " - $" + cartItems.get(i)[1]);

JButton removeButton = new JButton("Remove");

final int index = i; // Need a final variable to reference within ActionListener

removeButton.addActionListener(e -> {

cartItems.remove(index);

cartFrame.dispose(); // Close the cart frame

displayCart(); // Reopen the cart frame to refresh the cart items

});

cartItemPanel.add(cartItemLabel, BorderLayout.CENTER);

cartItemPanel.add(removeButton, BorderLayout.EAST);

cartPanel.add(cartItemPanel);

}

cartFrame.setVisible(true);

}

// Method to calculate total and display checkout info

private void checkout() {

double total = 0;

for (String[] item : cartItems) {

total += Double.parseDouble(item[1]);

}

JOptionPane.showMessageDialog(null, "Total amount to pay: $" + total);

}

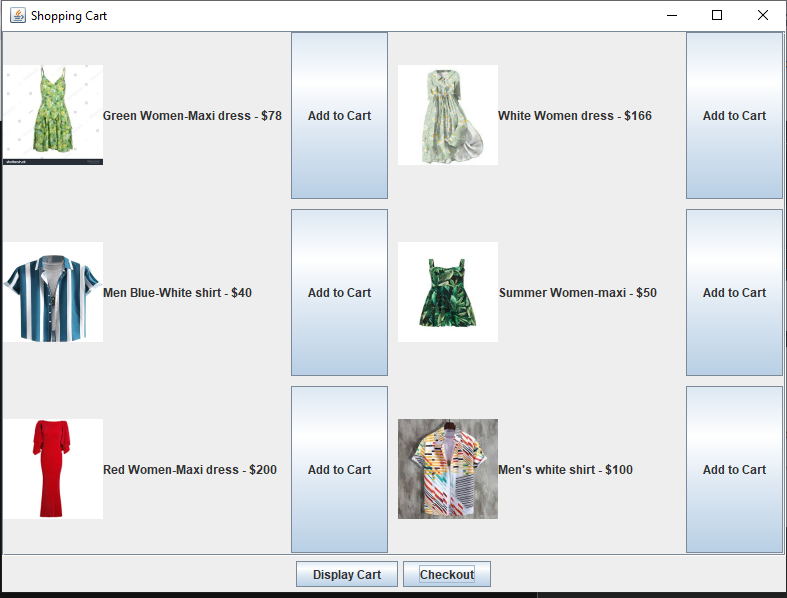
public static void main(String[] args) {

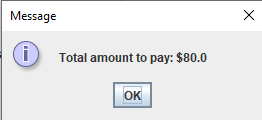
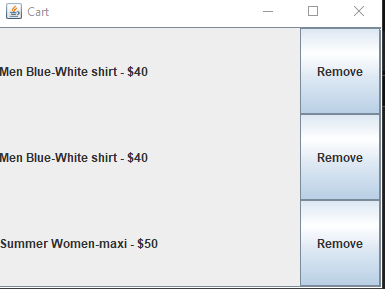
new CartSystem\_3045();

}

}

OUTPUT:





3. Write a program to create google account with details as shown in figure. Note :Use JPasswordField

CODE:

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.\*;

import java.util.regex.\*;

public class GoogleAccountCreation extends JFrame implements ActionListener {

    // Components of the form

    private Container container;

    private JLabel title, firstNameLabel, lastNameLabel, usernameLabel, passwordLabel, confirmPasswordLabel, phoneLabel, emailLabel, birthDateLabel, genderLabel, errorMessage;

    private JTextField firstNameField, lastNameField, usernameField, phoneField, emailField;

    private JPasswordField passwordField, confirmPasswordField;

    private JComboBox<String> monthCombo, dayCombo, yearCombo, genderCombo;

    private JComboBox<ImageIcon> countryCombo;

    private JButton submitButton, resetButton, cancelButton;

    // Constructor to initialize the form

    public GoogleAccountCreation() {

        setTitle("Google Account Creation");

        setBounds(300, 90, 600, 600);

        setDefaultCloseOperation(EXIT\_ON\_CLOSE);

        setResizable(false);

        container = getContentPane();

        container.setLayout(null);

        // Title

        title = new JLabel("Create your Google Account");

        title.setFont(new Font("Arial", Font.BOLD, 20));

        title.setSize(300, 30);

        title.setLocation(200, 30);

        container.add(title);

        // First Name

        firstNameLabel = new JLabel("First Name");

        firstNameLabel.setFont(new Font("Arial", Font.PLAIN, 15));

        firstNameLabel.setSize(100, 20);

        firstNameLabel.setLocation(50, 80);

        container.add(firstNameLabel);

        firstNameField = new JTextField();

        firstNameField.setFont(new Font("Arial", Font.PLAIN, 15));

        firstNameField.setSize(150, 20);

        firstNameField.setLocation(150, 80);

        container.add(firstNameField);

        // Last Name

        lastNameLabel = new JLabel("Last Name");

        lastNameLabel.setFont(new Font("Arial", Font.PLAIN, 15));

        lastNameLabel.setSize(100, 20);

        lastNameLabel.setLocation(50, 120);

        container.add(lastNameLabel);

        lastNameField = new JTextField();

        lastNameField.setFont(new Font("Arial", Font.PLAIN, 15));

        lastNameField.setSize(150, 20);

        lastNameField.setLocation(150, 120);

        container.add(lastNameField);

        // Username

        usernameLabel = new JLabel("Username");

        usernameLabel.setFont(new Font("Arial", Font.PLAIN, 15));

        usernameLabel.setSize(100, 20);

        usernameLabel.setLocation(50, 160);

        container.add(usernameLabel);

        usernameField = new JTextField();

        usernameField.setFont(new Font("Arial", Font.PLAIN, 15));

        usernameField.setSize(200, 20);

        usernameField.setLocation(150, 160);

        container.add(usernameField);

        // Password

        passwordLabel = new JLabel("Password");

        passwordLabel.setFont(new Font("Arial", Font.PLAIN, 15));

        passwordLabel.setSize(100, 20);

        passwordLabel.setLocation(50, 200);

        container.add(passwordLabel);

        passwordField = new JPasswordField();

        passwordField.setFont(new Font("Arial", Font.PLAIN, 15));

        passwordField.setSize(200, 20);

        passwordField.setLocation(150, 200);

        container.add(passwordField);

        // Confirm Password

        confirmPasswordLabel = new JLabel("Confirm");

        confirmPasswordLabel.setFont(new Font("Arial", Font.PLAIN, 15));

        confirmPasswordLabel.setSize(100, 20);

        confirmPasswordLabel.setLocation(50, 240);

        container.add(confirmPasswordLabel);

        confirmPasswordField = new JPasswordField();

        confirmPasswordField.setFont(new Font("Arial", Font.PLAIN, 15));

        confirmPasswordField.setSize(200, 20);

        confirmPasswordField.setLocation(150, 240);

        container.add(confirmPasswordField);

        // Phone Number with flag icons

        phoneLabel = new JLabel("Phone Number");

        phoneLabel.setFont(new Font("Arial", Font.PLAIN, 15));

        phoneLabel.setSize(120, 20);

        phoneLabel.setLocation(50, 280);

        container.add(phoneLabel);

        // Assuming flag images are stored in 'resources/flags' folder

        String[] countries = { "India", "USA" }; // More countries can be added

        ImageIcon indiaFlag = new ImageIcon("./flags/india.png");

        Image indiaF=indiaFlag.getImage().getScaledInstance(50, 25, DO\_NOTHING\_ON\_CLOSE);

        ImageIcon usaFlag = new ImageIcon("./flags/usa.png");

        Image usaF=usaFlag.getImage().getScaledInstance(50, 25, DO\_NOTHING\_ON\_CLOSE);

        indiaFlag=new ImageIcon(indiaF);

        usaFlag=new ImageIcon(usaF);

        ImageIcon[] flags={

            indiaFlag,usaFlag

        };

        countryCombo = new JComboBox<>(flags);

        countryCombo.setSize(70, 25);

        countryCombo.setLocation(150, 280);

        container.add(countryCombo);

        phoneField = new JTextField();

        phoneField.setFont(new Font("Arial", Font.PLAIN, 15));

        phoneField.setSize(200, 20);

        phoneField.setLocation(260, 280);

        container.add(phoneField);

        // Recovery Email

        emailLabel = new JLabel("Recovery Email");

        emailLabel.setFont(new Font("Arial", Font.PLAIN, 15));

        emailLabel.setSize(120, 20);

        emailLabel.setLocation(50, 320);

        container.add(emailLabel);

        emailField = new JTextField();

        emailField.setFont(new Font("Arial", Font.PLAIN, 15));

        emailField.setSize(200, 20);

        emailField.setLocation(150, 320);

        container.add(emailField);

        // Birth Date

        birthDateLabel = new JLabel("Your birthday");

        birthDateLabel.setFont(new Font("Arial", Font.PLAIN, 15));

        birthDateLabel.setSize(120, 20);

        birthDateLabel.setLocation(50, 360);

        container.add(birthDateLabel);

        String[] months = { "Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec" };

        String[] days = new String[31];

        String[] years = new String[50];

        for (int i = 1; i <= 31; i++) days[i-1] = String.valueOf(i);

        for (int i = 1975, j = 0; i <= 2024; i++, j++) years[j] = String.valueOf(i);

        monthCombo = new JComboBox<>(months);

        dayCombo = new JComboBox<>(days);

        yearCombo = new JComboBox<>(years);

        monthCombo.setSize(60, 20);

        dayCombo.setSize(50, 20);

        yearCombo.setSize(60, 20);

        monthCombo.setLocation(150, 360);

        dayCombo.setLocation(220, 360);

        yearCombo.setLocation(280, 360);

        container.add(monthCombo);

        container.add(dayCombo);

        container.add(yearCombo);

        // Gender

        genderLabel = new JLabel("Gender");

        genderLabel.setFont(new Font("Arial", Font.PLAIN, 15));

        genderLabel.setSize(100, 20);

        genderLabel.setLocation(50, 400);

        container.add(genderLabel);

        String[] genders = { "Male", "Female", "Other" };

        genderCombo = new JComboBox<>(genders);

        genderCombo.setSize(100, 20);

        genderCombo.setLocation(150, 400);

        container.add(genderCombo);

        // Buttons

        submitButton = new JButton("Create Account");

        submitButton.setFont(new Font("Arial", Font.PLAIN, 15));

        submitButton.setSize(150, 30);

        submitButton.setLocation(50, 450);

        submitButton.addActionListener(this);

        container.add(submitButton);

        resetButton = new JButton("Reset");

        resetButton.setFont(new Font("Arial", Font.PLAIN, 15));

        resetButton.setSize(100, 30);

        resetButton.setLocation(220, 450);

        resetButton.addActionListener(this);

        container.add(resetButton);

        cancelButton = new JButton("Cancel");

        cancelButton.setFont(new Font("Arial", Font.PLAIN, 15));

        cancelButton.setSize(100, 30);

        cancelButton.setLocation(350, 450);

        cancelButton.addActionListener(this);

        container.add(cancelButton);

    }

    // Action Performed event handling

    public void actionPerformed(ActionEvent e) {

        if (e.getSource() == submitButton) {

            try {

                validateForm();

                JOptionPane.showMessageDialog(this, "Account Created Successfully!");

            } catch (Exception ex) {

                showError(ex.getMessage());

            }

        } else if (e.getSource() == resetButton) {

            resetForm();

        } else if (e.getSource() == cancelButton) {

            System.exit(0);

        }

    }

    // Validate the form

    private void validateForm() throws Exception {

        String phoneRegex = "\\d{10}";

        Pattern pattern = Pattern.compile(phoneRegex);

        if (!pattern.matcher(phoneField.getText()).matches()) {

            throw new Exception("Invalid phone number");

        }

        // Add more validation logic as needed

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

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

        if (!password.equals(confirmPassword)) {

            throw new Exception("Passwords do not match");

        }

        // Check if all fields are filled

        if (firstNameField.getText().isEmpty() || lastNameField.getText().isEmpty() ||

            usernameField.getText().isEmpty() || phoneField.getText().isEmpty() ||

            emailField.getText().isEmpty()) {

            throw new Exception("All fields must be filled");

        }

    }

    // Reset form fields

    private void resetForm() {

        firstNameField.setText("");

        lastNameField.setText("");

        usernameField.setText("");

        passwordField.setText("");

        confirmPasswordField.setText("");

        phoneField.setText("");

        emailField.setText("");

        monthCombo.setSelectedIndex(0);

        dayCombo.setSelectedIndex(0);

        yearCombo.setSelectedIndex(0);

        genderCombo.setSelectedIndex(0);

    }

    // Show error messages

    private void showError(String message) {

        JOptionPane.showMessageDialog(this, message, "Validation Error", JOptionPane.ERROR\_MESSAGE);

    }

    // Main method

    public static void main(String[] args) {

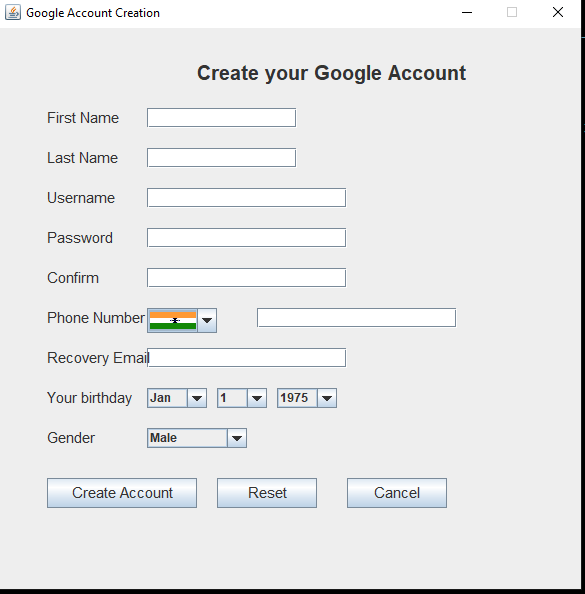
        GoogleAccountCreation form = new GoogleAccountCreation();

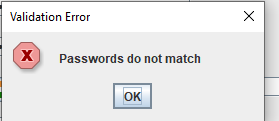
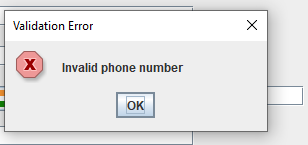
        form.setVisible(true);

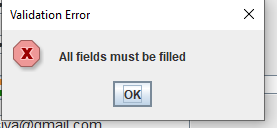
    }

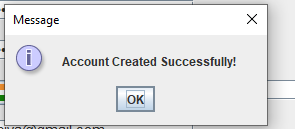
}

OUTPUT:







****