Login GUI

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
public class LoginGUI {
public static void main(String[] args) {
        // Create the login frame
        JFrame frame = new JFrame("Login");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
       // Create the login panel
        JPanel panel = new JPanel();
        panel.setLayout(new GridLayout(5, 3));
      // Add username label and text field
        JLabel usernameLabel = new JLabel("Username:");
        panel.add(usernameLabel);
        JTextField usernameField = new JTextField();
        panel.add(usernameField);
       // Add password label and text field
        JLabel passwordLabel = new JLabel("Password:");
        panel.add(passwordLabel);
        JPasswordField passwordField = new JPasswordField();
        passwordField.setEchoChar('*');
        panel.add(passwordField);
       // Create login button
        JButton loginButton = new JButton("Login");
        loginButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                String username = usernameField.getText();
                String password = new String(passwordField.getPassword());
             // Check if username and password are valid
```

```
                if (username.equals("admin") && password.equals("password")) {
                    // Login successful
                    JOptionPane.showMessageDialog(frame, "Login successful!");
                  // Display welcome message
                    String message = "Welcome, " + username + "!";
                    JOptionPane.showMessageDialog(frame, message);
                 // Close the login frame
                    frame.dispose();
                } else {
                    // Login failed
                    JOptionPane.showMessageDialog(frame, "Invalid username or password.");
                }
            }
        });
        panel.add(loginButton);
       // Add cancel button
        JButton cancelButton = new JButton("Cancel");
        cancelButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                // Close the login frame
                frame.dispose(); }
        });
        panel.add(cancelButton);
      // Add the panel to the frame and make it visible
        frame.add(panel);
        frame.pack();
        frame.setLocationRelativeTo(null);
       frame.setVisible(true);
    }
}
```

Animation Assignment

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

public class BouncingBallLoadingAnimation extends JFrame {
    private int ballX = 50;
    private int ballSpeed = 5;

    public BouncingBallLoadingAnimation() {
        setTitle("Bouncing Ball Loading Animation");
        setSize(400, 100);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
         Timer timer = new Timer(20, new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                moveBall();
                repaint(); // Request repaint to update the window
            }
        });
        timer.start();
    }
    private void moveBall() {
        ballX += ballSpeed;

        // Bounce off the walls
        if (ballX <= 0 || ballX >= getWidth() - 30) {
            ballSpeed = -ballSpeed;
        }
    }
    public void paint(Graphics g) {
        super.paint(g);
        // Draw the bouncing ball
        g.setColor(Color.BLUE);
        g.fillOval(ballX, 30, 30, 30);
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> {
            BouncingBallLoadingAnimation loadingAnimation = new BouncingBallLoadingAnimation();
            loadingAnimation.setVisible(true);
        });
    }
}
```

UDP Chat

UDPChatServer.java

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;


public class UDPChatServer {
    public static void main(String[] args) {
        try {
            DatagramSocket socket = new DatagramSocket(9876); // Server listens on port 9876


            System.out.println("UDP Chat Server is running...");


            while (true) {
                byte[] receiveData = new byte[1024];


                DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
                socket.receive(receivePacket);


                String clientMessage = new String(receivePacket.getData(), 0,
receivePacket.getLength());
                System.out.println("Client: " + clientMessage);
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

UDPChatClient.java

```
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Scanner;

public class UDPChatClient {
    public static void main(String[] args) {
        try {
            DatagramSocket socket = new DatagramSocket();
            InetAddress serverAddress = InetAddress.getByName("localhost"); // Server address
            int serverPort = 9876; // Server port
          Scanner scanner = new Scanner(System.in);
           while (true) {
                System.out.print("You: ");
                String message = scanner.nextLine();
              byte[] sendData = message.getBytes();
                DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, serverAddress, serverPort);
                socket.send(sendPacket);
                if (message.equalsIgnoreCase("exit")) {
                    System.out.println("Closing chat client...");
                    break;
                }
            }
       socket.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```