

1. Create A Simple User Registration Form Using Awt In Java Which Has The Fields likes FirstName, Last-Name, Date Of Birth, Gender – Male Or Female(Check box Or Radio Button), Place ,Contact Number and Submit button

```
package com.challengeawtswing;
import java.awt.*;
import java.awt.event.*;
import javax.swing.JOptionPane;

public class UserRegistrationForm extends Frame implements ActionListener {
    private Label firstNameLabel, lastNameLabel, dobLabel, genderLabel,
    placeLabel, contactLabel;
    private TextField firstNameField, lastNameField, dobField, placeField,
    contactField;
    private CheckboxGroup genderCheckboxGroup;
    private Checkbox maleCheckbox, femaleCheckbox;
    private Button submitButton;

    public UserRegistrationForm() {
        setLayout(new GridLayout(8, 2)); // 8 rows, 2 columns

        firstNameLabel = new Label("First Name:");
        lastNameLabel = new Label("Last Name:");
        dobLabel = new Label("Date of Birth (YYYY-MM-DD):");
        genderLabel = new Label("Gender:");
        placeLabel = new Label("Place:");
        contactLabel = new Label("Contact Number:");

        firstNameField = new TextField(20);
        lastNameField = new TextField(20);
        dobField = new TextField(20);
        placeField = new TextField(20);
        contactField = new TextField(20);

        genderCheckboxGroup = new CheckboxGroup();
        maleCheckbox = new Checkbox("Male", genderCheckboxGroup, false);
        femaleCheckbox = new Checkbox("Female", genderCheckboxGroup,
false);

        submitButton = new Button("Submit");
        submitButton.addActionListener(this);

        add(firstNameLabel);
        add(firstNameField);
        add(lastNameLabel);
        add(lastNameField);
        add(dobLabel);
        add(dobField);
        add(genderLabel);
        add(maleCheckbox);
        add(new Label("")); // Empty label for spacing
        add(femaleCheckbox);
        add(placeLabel);
        add(placeField);
        add(contactLabel);
        add(contactField);
        add(new Label("")); // Empty label for spacing
        add(submitButton);
    }
}
```

```

        setTitle("User Registration Form");
        setSize(400, 300);
        setVisible(true);

        addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent e) {
                dispose();
            }
        });
    }

    public void actionPerformed(ActionEvent e) {
        if (e.getSource() == submitButton) {
            String firstName = firstNameField.getText();
            String lastName = lastNameField.getText();
            String dob = dobField.getText();
            String gender = (maleCheckbox.getState() ? "Male" : "Female");
            String place = placeField.getText();
            String contact = contactField.getText();

            // Here you can write your logic to handle the form data
            // For example, saving to a database or displaying a message
            dialog
            // For now, let's just display a message dialog
            String message = "Registration Successful!\nThank you, " +
            firstName + " " + lastName + "!";
            JOptionPane.showMessageDialog(this, message, "Registration
            Successful", JOptionPane.INFORMATION_MESSAGE);

            // Clear fields after submission
            firstNameField.setText("");
            lastNameField.setText("");
            dobField.setText("");
            genderCheckboxGroup.setSelectedCheckbox(null); // Clear gender
            selection
            placeField.setText("");
            contactField.setText("");
        }
    }

    public static void main(String[] args) {
        new UserRegistrationForm();
    }
}

```

Output:

User Registration Form

First Name:

Last Name:

Date of Birth (YYYY-MM-DD):

Gender:

☐ Male

☐ Female

Place:

Contact Number:

Submit

2. How To Make A Simple Calculator Using AWT & Swing In Java

```
package com.challengeawtswing;
import javax.swing.*.*;
import java.awt.*.*;
import java.awt.event.*;

public class SimpleCalculator extends JFrame implements ActionListener {
    private JTextField display;
    private JButton[] buttons;
    private String[] buttonLabels = {
        "7", "8", "9", "/", // Row 1
        "4", "5", "6", "*", // Row 2
        "1", "2", "3", "-", // Row 3
        "0", ".", "=", "+" // Row 4
    };

    private double num1, num2;
    private char operator;

    public SimpleCalculator() {
        setTitle("Simple Calculator");
        setSize(300, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        display = new JTextField(10);
        display.setEditable(false);
        add(display, BorderLayout.NORTH);

        JPanel buttonPanel = new JPanel();
        buttonPanel.setLayout(new GridLayout(4, 4, 5, 5));

        buttons = new JButton[buttonLabels.length];
        for (int i = 0; i < buttonLabels.length; i++) {
            buttons[i] = new JButton(buttonLabels[i]);
            buttons[i].addActionListener(this);
        }
    }
}
```

```

        buttonPanel.add(buttons[i]);
    }

    add(buttonPanel, BorderLayout.CENTER);

    setVisible(true);
}

public void actionPerformed(ActionEvent e) {
    String command = e.getActionCommand();

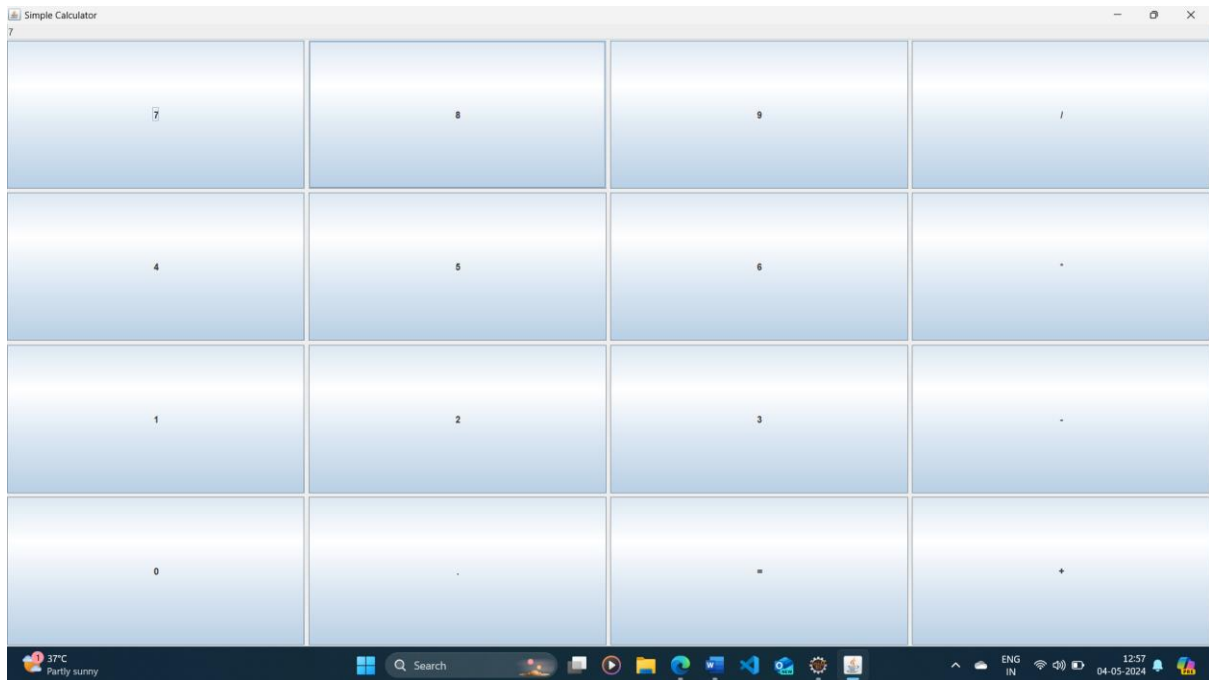
    if (Character.isDigit(command.charAt(0)) || command.equals(".")) {
        display.setText(display.getText() + command);
    } else if (command.equals("=")) {
        num2 = Double.parseDouble(display.getText());
        double result = calculate(num1, num2, operator);
        display.setText(String.valueOf(result));
    } else {
        num1 = Double.parseDouble(display.getText());
        operator = command.charAt(0);
        display.setText("");
    }
}

private double calculate(double num1, double num2, char operator) {
    switch (operator) {
        case '+':
            return num1 + num2;
        case '-':
            return num1 - num2;
        case '*':
            return num1 * num2;
        case '/':
            if (num2 != 0)
                return num1 / num2;
            else
                return 0;
        default:
            return 0;
    }
}

public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
        public void run() {
            new SimpleCalculator();
        }
    });
}
}

```

Output:



3. How To Create A Simple Notepad In Java Using Swing application

```
package com.challengeawtswing;

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

public class SimpleNotepad extends JFrame {
    private JTextArea textArea;
    private JFileChooser fileChooser;

    public SimpleNotepad() {
        setTitle("Simple Notepad");
        setSize(800, 600);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        textArea = new JTextArea();
        JScrollPane scrollPane = new JScrollPane(textArea);
        add(scrollPane, BorderLayout.CENTER);

        JMenuBar menuBar = new JMenuBar();
        setJMenuBar(menuBar);

        JMenu fileMenu = new JMenu("File");
        menuBar.add(fileMenu);

        JMenuItem newMenuItem = new JMenuItem("New");
        fileMenu.add(newMenuItem);
        newMenuItem.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
```

```

        textArea.setText(""); // Clear the text area
    }
});

JMenuItem openMenuItem = new JMenuItem("Open");
fileMenu.add(openMenuItem);
openMenuItem.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        openFile(); // Open a file
    }
});

JMenuItem saveMenuItem = new JMenuItem("Save");
fileMenu.add(saveMenuItem);
saveMenuItem.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        saveFile(); // Save the current text to a file
    }
});

JMenuItem exitMenuItem = new JMenuItem("Exit");
fileMenu.add(exitMenuItem);
exitMenuItem.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        System.exit(0); // Exit the application
    }
});

setVisible(true);
}

private void openFile() {
    if (fileChooser == null) {
        fileChooser = new JFileChooser();
    }
    int returnVal = fileChooser.showOpenDialog(this);
    if (returnVal == JFileChooser.APPROVE_OPTION) {
        File file = fileChooser.getSelectedFile();
        try {
            FileReader fileReader = new FileReader(file);
            BufferedReader reader = new BufferedReader(fileReader);
            textArea.read(reader, null);
            reader.close();
            fileReader.close();
        } catch (IOException ex) {
            ex.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error opening file",
"Error", JOptionPane.ERROR_MESSAGE);
        }
    }
}

private void saveFile() {
    if (fileChooser == null) {
        fileChooser = new JFileChooser();
    }
    int returnVal = fileChooser.showSaveDialog(this);
    if (returnVal == JFileChooser.APPROVE_OPTION) {
        File file = fileChooser.getSelectedFile();
        try {
            FileWriter fileWriter = new FileWriter(file);

```

Output:



```

setSize(800, 600);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

customers = new ArrayList<>();

// Initialize panels
loginPanel = new JPanel();
homePanel = new JPanel();
createAccountPanel = new JPanel();
viewCustomersPanel = new JPanel();

setupLoginPanel();
setupHomePanel();
setupCreateAccountPanel();
setupViewCustomersPanel();

add(loginPanel);
setVisible(true);
}

private void setupLoginPanel() {
    loginPanel.setLayout(new GridLayout(3, 2));
    JLabel usernameLabel = new JLabel("Username:");
    usernameField = new JTextField();
    JLabel passwordLabel = new JLabel("Password:");
    passwordField = new JPasswordField();
    JButton loginButton = new JButton("Login");

    loginButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            String username = usernameField.getText();
            String password = new String(passwordField.getPassword());

            if (username.equals("admin") && password.equals("admin")) {
                // Login successful, switch to home panel
                getContentPane().removeAll();
                add(homePanel);
                revalidate();
            } else {
                JOptionPane.showMessageDialog(loginPanel, "Invalid
username or password", "Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    });

    loginPanel.add(usernameLabel);
    loginPanel.add(usernameField);
    loginPanel.add(passwordLabel);
    loginPanel.add(passwordField);
    loginPanel.add(new JLabel()); // Empty label for spacing
    loginPanel.add(loginButton);
}

private void setupHomePanel() {
    homePanel.setLayout(new GridLayout(3, 1));

    JButton createAccountButton = new JButton("Create Account");
    createAccountButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            getContentPane().removeAll();
            add(createAccountPanel);
        }
    });
}

```



```

        revalidate();
    }
});

JButton viewCustomersButton = new JButton("View All Customers");
viewCustomersButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        updateCustomerList();
        getContentPane().removeAll();
        add(viewCustomersPanel);
        revalidate();
    }
});

JButton logoutButton = new JButton("Logout");
logoutButton.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        getContentPane().removeAll();
        add(loginPanel);
        revalidate();
    }
});

homePanel.add(createAccountButton);
homePanel.add(viewCustomersButton);
homePanel.add(logoutButton);
}

private void setupCreateAccountPanel() {
    createAccountPanel.setLayout(new GridLayout(4, 2));
    JLabel nameLabel = new JLabel("Name:");
    JTextField nameField = new JTextField();
    JLabel accountNumberLabel = new JLabel("Account Number:");
    JTextField accountNumberField = new JTextField();
    JLabel initialBalanceLabel = new JLabel("Initial Balance:");
    JTextField initialBalanceField = new JTextField();
    JButton createButton = new JButton("Create");

    createButton.addActionListener(new ActionListener() {
        public void actionPerformed(ActionEvent e) {
            String name = nameField.getText();
            String accountNumber = accountNumberField.getText();
            double initialBalance =
Double.parseDouble(initialBalanceField.getText());

            Customer customer = new Customer(name, accountNumber,
initialBalance);
            customers.add(customer);

            JOptionPane.showMessageDialog(createAccountPanel, "Account
created successfully", "Success", JOptionPane.INFORMATION_MESSAGE);
            clearFields(nameField, accountNumberField,
initialBalanceField);
        }
    });

    createAccountPanel.add(nameLabel);
    createAccountPanel.add(nameField);
    createAccountPanel.add(accountNumberLabel);
    createAccountPanel.add(accountNumberField);
    createAccountPanel.add(initialBalanceLabel);

```

```

        createAccountPanel.add(initialBalanceField);
        createAccountPanel.add(new JLabel()); // Empty label for spacing
        createAccountPanel.add(createButton);
    }

    private void setupViewCustomersPanel() {
        viewCustomersPanel.setLayout(new BorderLayout());
        JLabel titleLabel = new JLabel("All Customers:");
        customerListArea = new JTextArea(20, 40);
        customerListArea.setEditable(false);
        JScrollPane scrollPane = new JScrollPane(customerListArea);

        viewCustomersPanel.add(titleLabel, BorderLayout.NORTH);
        viewCustomersPanel.add(scrollPane, BorderLayout.CENTER);
    }

    private void updateCustomerList() {
        StringBuilder sb = new StringBuilder();
        for (Customer customer : customers) {
            sb.append(customer.toString()).append("\n");
        }
        customerListArea.setText(sb.toString());
    }

    private void clearFields(JTextField... fields) {
        for (JTextField field : fields) {
            field.setText("");
        }
    }

    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new BankingApplication();
            }
        });
    }
}

class Customer {
    private String name;
    private String accountNumber;
    private double balance;

    public Customer(String name, String accountNumber, double balance) {
        this.name = name;
        this.accountNumber = accountNumber;
        this.balance = balance;
    }

    public String getName() {
        return name;
    }

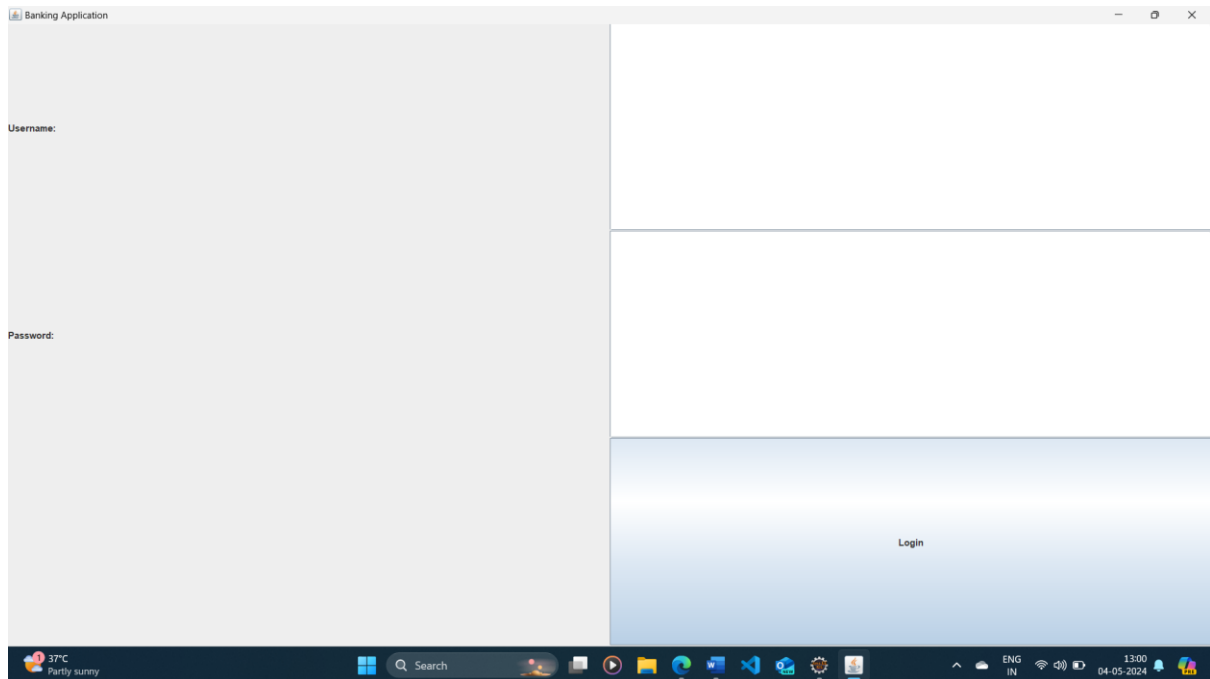
    public String getAccountNumber() {
        return accountNumber;
    }

    public double getBalance() {
        return balance;
    }
}

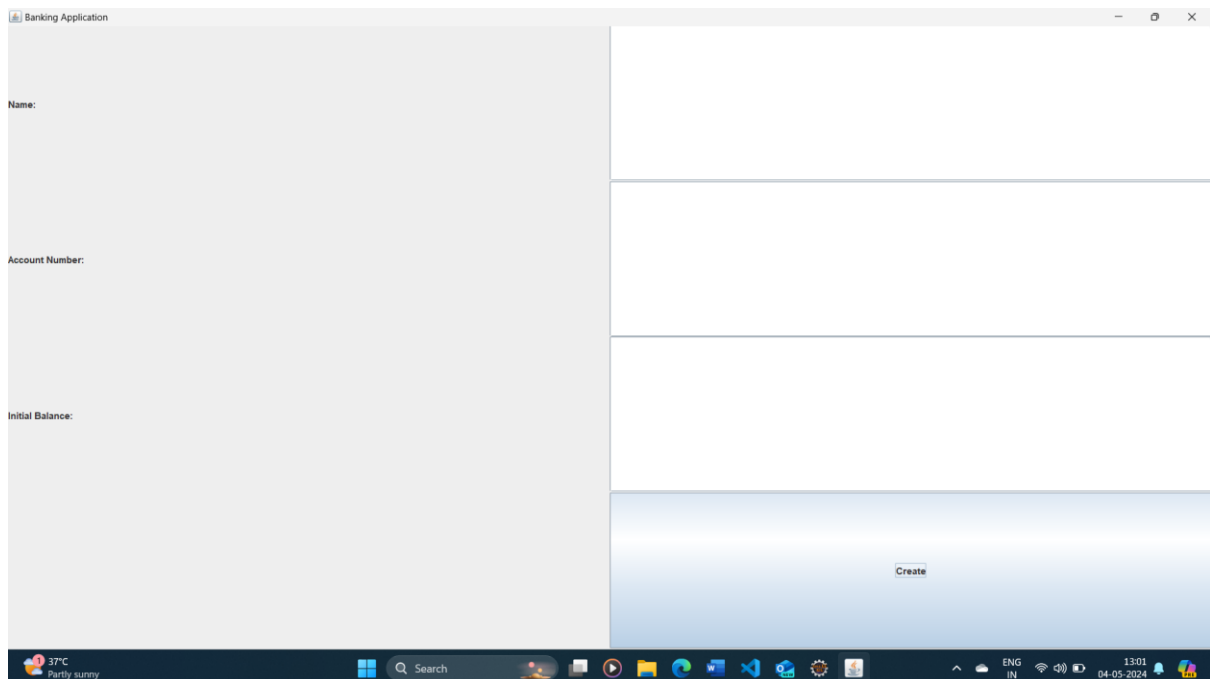
```

```
@Override
public String toString() {
    return "Name: " + name + ", Account Number: " + accountNumber + ",
Balance: $" + balance;
}
}
```

Output:



A screenshot of a Java Swing window titled "Banking Application". The window is divided into two main sections. The left section is a light gray panel with labels "Username:" and "Password:". The right section is a white panel with two empty text input fields stacked vertically, followed by a blue button labeled "Login". The Windows taskbar is visible at the bottom, showing the date and time as 13:00 on 04-05-2024.



A screenshot of a Java Swing window titled "Banking Application". The window is divided into two main sections. The left section is a light gray panel with labels "Name:", "Account Number:", and "Initial Balance:". The right section is a white panel with three empty text input fields stacked vertically, followed by a blue button labeled "Create". The Windows taskbar is visible at the bottom, showing the date and time as 13:01 on 04-05-2024.