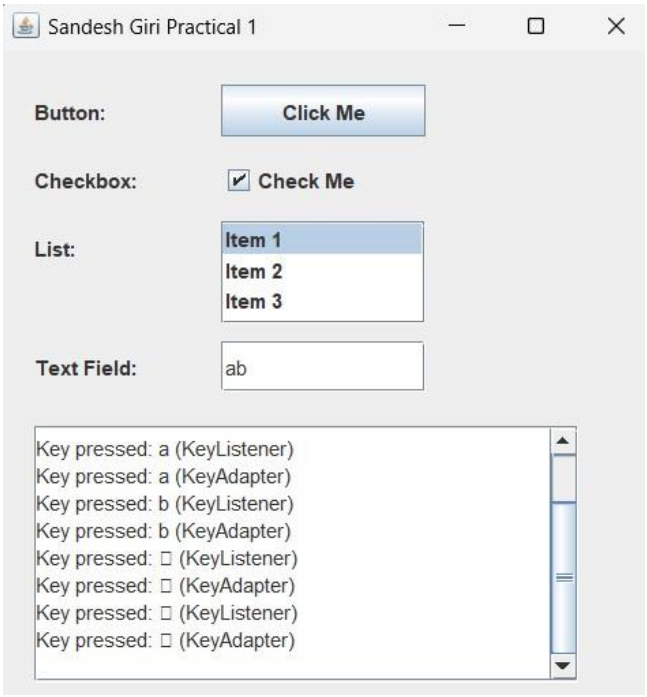
1. Write a program to handle action event, item event, mouse event (both adapter and elegation method), key event (both adapter and delegation method), list selection event.

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import javax.swing.event.*;
public class Practical1 extends JFrame {
   JButton button;
   JCheckBox checkBox;
   JList<String> list;
   JTextField textField;
   JTextArea displayArea;

   public Practical1() {
      setTitle("Sandesh Giri Practical 1");
      setLayout(null);
      button = new JButton("Click Me");
      checkBox = new JCheckBox("Check Me");
      list = new JList<>(new String[]{"Item 1", "Item 2", "Item3"});

      textField = new JTextField(20);
      displayArea = new JTextArea();
      displayArea.setEditable(false);
      JScrollPane displayScrollPane = new JScrollPane(displayArea);
      JLabel buttonLabel = new JLabel("Button:");
      JLabel checkBoxLabel = new JLabel("Checkbox:");
      JLabel listLabel = new JLabel("List:");
      JLabel textFieldLabel = new JLabel("Text Field:");

      buttonLabel.setBounds(20, 20, 100, 30);
      button.setBounds(130, 20, 120, 30);

      checkBoxLabel.setBounds(20, 60, 100, 30);
      checkBox.setBounds(130, 60, 120, 30);

      listLabel.setBounds(20, 100, 100, 30);
      JScrollPane listScrollPane = new JScrollPane(list);
      listScrollPane.setBounds(130, 100, 120, 60);

      textFieldLabel.setBounds(20, 170, 100, 30);
      textField.setBounds(130, 170, 120, 30);
      displayScrollPane.setBounds(20, 220, 320, 150);
      add(buttonLabel);
      add(button);
      add(checkBoxLabel);
```

```
        add(checkBox);
        add(listLabel);
        add(listScrollPane);
        add(textFieldLabel);
        add(textField);
        add(displayScrollPane);
        button.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                displayArea.append("Button clicked\n");
            }
        });
        checkBox.addItemListener(new ItemListener() {
            public void itemStateChanged(ItemEvent e) {
                displayArea.append("Checkbox state changed\n");
            }
        });
        list.addListSelectionListener(new ListSelectionListener() {
            public void valueChanged(ListSelectionEvent e) {
                if (!e.getValueIsAdjusting()) {
                    displayArea.append("List item selected: " + list.getSelectedValue() + "\n");
                }
            }
        });
        textField.addKeyListener(new KeyListener() {
            public void keyPressed(KeyEvent e) {
                displayArea.append("Key pressed: " + e.getKeyChar() + " (KeyListener)\n");
            }
            public void keyReleased(KeyEvent e) {}
            public void keyTyped(KeyEvent e) {}
        });

        textField.addKeyListener(new KeyAdapter() {
            public void keyPressed(KeyEvent e) {
                displayArea.append("Key pressed: " + e.getKeyChar() + " (KeyAdapter)\n");
            }
        });
        addMouseListener(new MouseListener() {
            public void mouseClicked(MouseEvent e) {
                displayArea.append("Mouse clicked at: (" + e.getX() + ", " + e.getY() + ")
(MouseListener)\n");
            }
            public void mousePressed(MouseEvent e) {}
            public void mouseReleased(MouseEvent e) {}
            public void mouseEntered(MouseEvent e) {}
            public void mouseExited(MouseEvent e) {}
        });
```

```
        addMouseListener(new MouseAdapter() {
            public void mouseClicked(MouseEvent e) {
                displayArea.append("Mouse clicked at: (" + e.getX() + ", " + e.getY() + ")
(MouseAdapter)\n");
            }
        });
        setSize(400, 450);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public static void main(String[] args) {
        new Practical1();
    }
}
```

Output :

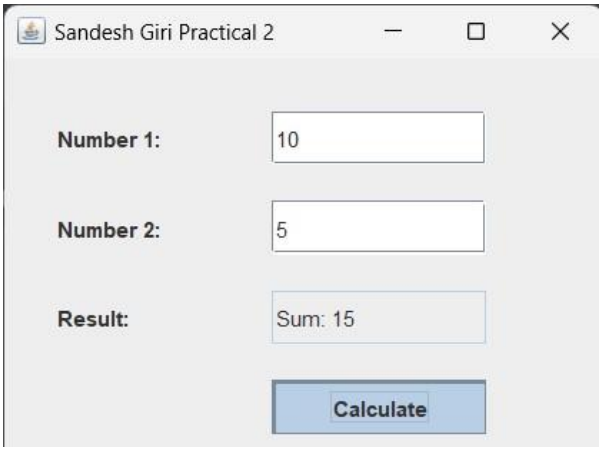


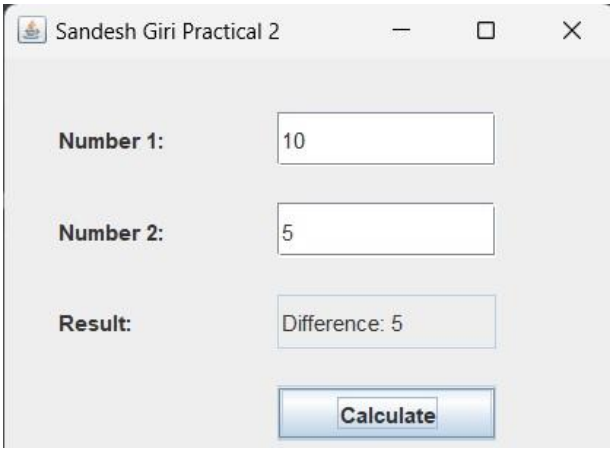
2. Write a program to find the sum and difference of two number. Use two textfield to take input and another one text field to display result. Display sum of two number if mouse is clicked and display difference if mouse is released.

```
import javax.swing.*;
import java.awt.event.*;
public class Practical2 extends JFrame {
   JTextField number1Field, number2Field, resultField;
   JButton calculateButton;
   public Practical2() {
      setTitle("Sandesh Giri Practical 2");
      setLayout(null);
      JLabel label1 = new JLabel("Number 1:");
      JLabel label2 = new JLabel("Number 2:");
      JLabel resultLabel = new JLabel("Result:");
      calculateButton = new JButton("Calculate");

      number1Field = new JTextField(10);
      number2Field = new JTextField(10);
      resultField = new JTextField(10);
      resultField.setEditable(false);

      label1.setBounds(30, 30, 100, 30);
      number1Field.setBounds(150, 30, 120, 30);
      label2.setBounds(30, 80, 100, 30);
      number2Field.setBounds(150, 80, 120, 30);
      resultLabel.setBounds(30, 130, 100, 30);
      resultField.setBounds(150, 130, 120, 30);
      calculateButton.setBounds(150, 180, 120, 30);

      add(label1);
      add(number1Field);
      add(label2);
      add(number2Field);
      add(resultLabel);
      add(resultField);
      add(calculateButton);

      calculateButton.addMouseListener(new MouseAdapter() {
         public void mousePressed(MouseEvent e) {
            calculateSum(); }
         public void mouseReleased(MouseEvent e) {
            calculateDifference(); }
```

```
        });
        setSize(350, 300);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true); }
    public void calculateSum() {
        try {
            int num1 = Integer.parseInt(number1Field.getText());
            int num2 = Integer.parseInt(number2Field.getText());
            int sum = num1 + num2;
            resultField.setText("Sum: " + sum);
        } catch (NumberFormatException e) {
            resultField.setText("Invalid Input"); }
    }
    public void calculateDifference() {
        try {
            int num1 = Integer.parseInt(number1Field.getText());
            int num2 = Integer.parseInt(number2Field.getText());
            int diff = num1 - num2;
            resultField.setText("Difference: " + diff);
        } catch (NumberFormatException e) {
            resultField.setText("Invalid Input"); }
    }
    public static void main(String[] args) {
        new Practical2(); }
}
```

Output :





3. Create two four button and one text field. If 1st button is clicked display first button is clicked in text field and same for all other button.

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

public class Practical3 extends JFrame {
   JTextField resultField;
   JButton button1, button2, button3, button4;

   public Practical3() {
      setTitle("Sandesh Giri Practical 3");
      setLayout(null);

      resultField = new JTextField(20);
      button1 = new JButton("Button 1");
      button2 = new JButton("Button 2");
      button3 = new JButton("Button 3");
      button4 = new JButton("Button 4");

      resultField.setBounds(50, 30, 200, 30);
      button1.setBounds(50, 80, 100, 30);
      button2.setBounds(160, 80, 100, 30);
      button3.setBounds(50, 130, 100, 30);
      button4.setBounds(160, 130, 100, 30);

      add(resultField);
      add(button1);
      add(button2);
      add(button3);
      add(button4);

      button1.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
            resultField.setText("First button is clicked");
         }
      });

      button2.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
            resultField.setText("Second button is clicked");
         }
      });
```

```
        button3.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                resultField.setText("Third button is clicked");
            }
        });

        button4.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                resultField.setText("Fourth button is clicked");
            }
        });

        setSize(320, 250);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }

    public static void main(String[] args) {
        new Practical3();
    }
}
```

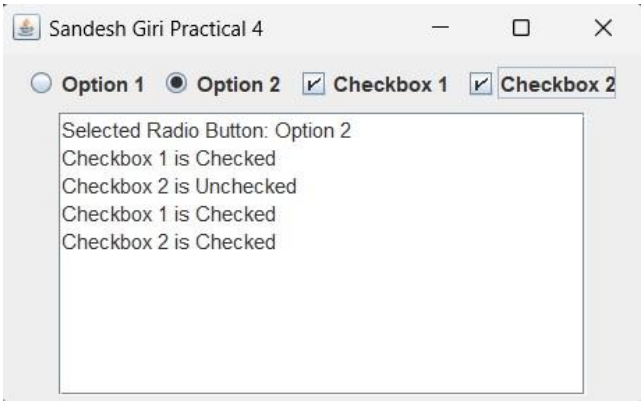
Output :

4. Create radio button and checkbox of your choice and handle the event generated by both.

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

public class Practical4 extends JFrame {
   private JRadioButton radioButton1;
   private JRadioButton radioButton2;
   private JCheckBox checkBox1;
   private JCheckBox checkBox2;
   private JTextArea outputArea;
   public Practical4() {
      setTitle("Sandesh Giri Practical 4");
      setSize(400, 300);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setLayout(new FlowLayout());

      radioButton1 = new JRadioButton("Option 1");
      radioButton2 = new JRadioButton("Option 2");

      ButtonGroup group = new ButtonGroup();
      group.add(radioButton1);
      group.add(radioButton2);

      checkBox1 = new JCheckBox("Checkbox 1");
      checkBox2 = new JCheckBox("Checkbox 2");

      outputArea = new JTextArea(10, 30);
      outputArea.setEditable(false);

      radioButton1.addActionListener(new RadioButtonListener());
      radioButton2.addActionListener(new RadioButtonListener());
      checkBox1.addActionListener(new CheckBoxListener());
      checkBox2.addActionListener(new CheckBoxListener());
      add(radioButton1);
      add(radioButton2);
      add(checkBox1);
      add(checkBox2);
      add(new JScrollPane(outputArea));
      setVisible(true);
   }
   private class RadioButtonListener implements ActionListener {
```

```
        @Override
        public void actionPerformed(ActionEvent e) {
            JRadioButton selectedButton = (JRadioButton) e.getSource();
            outputArea.setText("Selected Radio Button: " + selectedButton.getText() + "\n");
            updateCheckboxStates();
        }
    }
    private class CheckBoxListener implements ActionListener {
        @Override
        public void actionPerformed(ActionEvent e) {
            updateCheckboxStates();
        }
    }
    private void updateCheckboxStates() {
        StringBuilder sb = new StringBuilder();
        sb.append("Checkbox 1 is ").append(checkBox1.isSelected() ? "Checked" :
"Unchecked").append("\n");
        sb.append("Checkbox 2 is ").append(checkBox2.isSelected() ? "Checked" :
"Unchecked");
        outputArea.append(sb.toString() + "\n");
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(() -> new Practical4());
    }
}
```

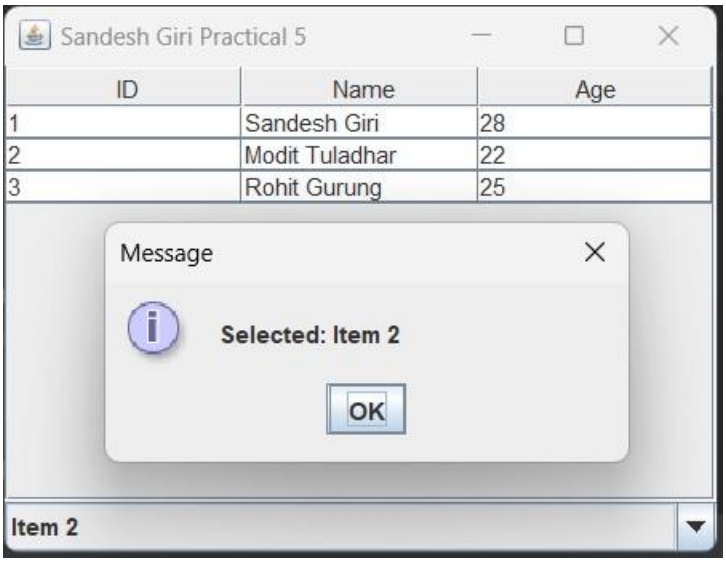
Output :



5. How Jtable and JComboBox is created? Show with program.

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

public class Practical5 {

   public static void main(String[] args) {
      JFrame frame = new JFrame("Sandesh Giri Practical 5");
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setSize(300, 300);

      String[] columnNames = {"ID", "Name", "Age"};
      Object[][] data = {
            {1, "Sandesh Giri", 28},
            {2, "Modit Tuladhar", 22},
            {3, "Rohit Gurung", 25}
      };

      JTable table = new JTable(data, columnNames);
      JScrollPane tableScrollPane = new JScrollPane(table);

      String[] comboBoxItems = {"Item 1", "Item 2", "Item 3"};
      JComboBox<String> comboBox = new JComboBox<>(comboBoxItems);
      comboBox.addActionListener(new ActionListener() {

         @Override
         public void actionPerformed(ActionEvent e) {
            String selectedItem = (String) comboBox.getSelectedItem();
            JOptionPane.showMessageDialog(frame, "Selected: " + selectedItem);
         }
      });

      JPanel panel = new JPanel();
      panel.setLayout(new BorderLayout());
      panel.add(tableScrollPane, BorderLayout.CENTER);
      panel.add(comboBox, BorderLayout.SOUTH);
      frame.add(panel);
      frame.setVisible(true);
   }
}
```

Output :


Sandesh Giri Practical 5
ID
Name
Age
1
Sandesh Giri
28
2
Modit Tuladhar
22
3
Rohit Gurung
25
Message
Selected: Item 2
OK
Item 2

6. Create one complete form using swing (username, password, repassword, radio button, check box, submit and reset button). validate the form (check for emptiness, password and repassword should be equal) and insert the data into database. Assume your own database and table.

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

public class Practical6 {
   private static final String DB_URL = "jdbc:mysql://localhost:3306/testdb";
   private static final String DB_USER = "root";
   private static final String DB_PASSWORD = "";

   public static void main(String[] args) {
      JFrame frame = new JFrame("Sandesh Giri Practical 6");
      frame.setSize(400, 300);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setLayout(null);

      JLabel lblUsername = new JLabel("Username:");
      lblUsername.setBounds(10, 20, 100, 25);
      JTextField txtUsername = new JTextField();
      txtUsername.setBounds(120, 20, 200, 25);

      JLabel lblPassword = new JLabel("Password:");
      lblPassword.setBounds(10, 60, 100, 25);
      JPasswordField txtPassword = new JPasswordField();
      txtPassword.setBounds(120, 60, 200, 25);

      JLabel lblRepassword = new JLabel("Re-password");
      lblRepassword.setBounds(10, 100, 100, 25);
      JPasswordField txtRepassword = new JPasswordField();
      txtRepassword.setBounds(120, 100, 200, 25);

      JLabel lblGender = new JLabel("Gender:");
      lblGender.setBounds(10, 140, 100, 25);
      JRadioButton rbMale = new JRadioButton("Male");
      rbMale.setBounds(120, 140, 70, 25);
```

```
        JRadioButton rbFemale = new JRadioButton("Female");
        rbFemale.setBounds(190, 140, 80, 25);
        ButtonGroup genderGroup = new ButtonGroup();
        genderGroup.add(rbMale);
        genderGroup.add(rbFemale);

        JCheckBox chkAgree = new JCheckBox("Agree to terms");
        chkAgree.setBounds(10, 180, 150, 25);

        JButton btnSubmit = new JButton("Submit");
        btnSubmit.setBounds(50, 220, 100, 30);
        JButton btnReset = new JButton("Reset");
        btnReset.setBounds(200, 220, 100, 30);

        frame.add(lblUsername);
        frame.add(txtUsername);
        frame.add(lblPassword);
        frame.add(txtPassword);
        frame.add(lblRepassword);
        frame.add(txtRepassword);
        frame.add(lblGender);
        frame.add(rbMale);
        frame.add(rbFemale);
        frame.add(chkAgree);
        frame.add(btnSubmit);
        frame.add(btnReset);

        btnSubmit.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String username = txtUsername.getText();
                String password = new String(txtPassword.getPassword());
                String repassword = new String(txtRepassword.getPassword());
                String gender = rbMale.isSelected() ? "Male" : rbFemale.isSelected() ? "Female" :
"";
                boolean agreeTerms = chkAgree.isSelected();

                if (username.isEmpty() || password.isEmpty() || repassword.isEmpty()) {
                    JOptionPane.showMessageDialog(frame, "Please fill in all fields.", "Validation
Error", JOptionPane.ERROR_MESSAGE);
                    return;
                }
```

```
            if (!password.equals(repassword)) {
                JOptionPane.showMessageDialog(frame, "Passwords do not match.", "Validation
Error", JOptionPane.ERROR_MESSAGE);
                return;
            }

            if (gender.isEmpty()) {
                JOptionPane.showMessageDialog(frame, "Please select your gender.",
"Validation Error", JOptionPane.ERROR_MESSAGE);
                return;
            }

            try (Connection conn = DriverManager.getConnection(DB_URL, DB_USER,
DB_PASSWORD)) {
                String sql = "INSERT INTO users (username, password, gender, agree_terms)
VALUES (?, ?, ?, ?)";
                try (PreparedStatement stmt = conn.prepareStatement(sql)) {
                    stmt.setString(1, username);
                    stmt.setString(2, password);
                    stmt.setString(3, gender);
                    stmt.setBoolean(4, agreeTerms);
                    stmt.executeUpdate();
                    JOptionPane.showMessageDialog(frame, "Registration successful.",
"Success", JOptionPane.INFORMATION_MESSAGE);
                }
            } catch (SQLException ex) {
                JOptionPane.showMessageDialog(frame, "Database error: " + ex.getMessage(),
"Error", JOptionPane.ERROR_MESSAGE);
            }
        }
    });

    btnReset.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {
            txtUsername.setText("");
            txtPassword.setText("");
            txtRepassword.setText("");
            genderGroup.clearSelection();
            chkAgree.setSelected(false);
        }
    });
    frame.setVisible(true);
```

```
    }
}
```

Output :

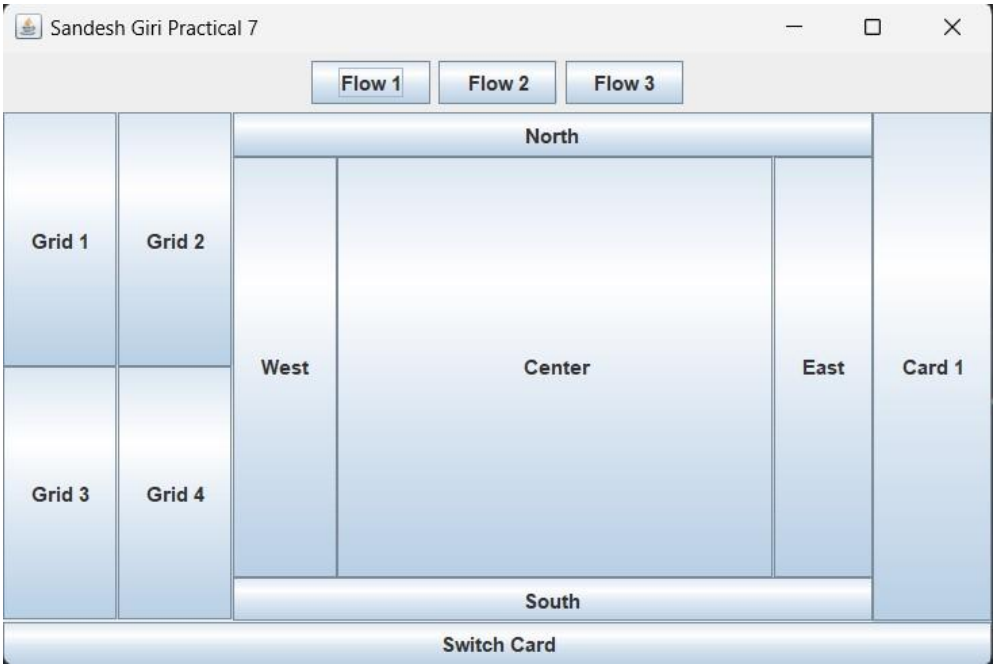

Sandesh Giri Practical 6
Username:
Sandesh Giri
Password:
Re-password
Gender:
Male
Female
Agree to terms
Submit
Reset
Success
Registration successful.
OK

7. Write a program to demonstrate different kinds of layout manager (null, flow, grid, border, card).

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

public class Practical7 {
   public static void main(String[] args) {
      JFrame frame = new JFrame("Sandesh Giri Practical 7");
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setSize(600, 400);
      frame.setLayout(new BorderLayout());

      JPanel flowPanel = new JPanel(new FlowLayout());
      flowPanel.add(new JButton("Flow 1"));
      flowPanel.add(new JButton("Flow 2"));
      flowPanel.add(new JButton("Flow 3"));

      JPanel gridPanel = new JPanel(new GridLayout(2, 2));
      gridPanel.add(new JButton("Grid 1"));
      gridPanel.add(new JButton("Grid 2"));
      gridPanel.add(new JButton("Grid 3"));
      gridPanel.add(new JButton("Grid 4"));

      JPanel borderPanel = new JPanel(new BorderLayout());
      borderPanel.add(new JButton("North"), BorderLayout.NORTH);
      borderPanel.add(new JButton("South"), BorderLayout.SOUTH);
      borderPanel.add(new JButton("East"), BorderLayout.EAST);
      borderPanel.add(new JButton("West"), BorderLayout.WEST);
      borderPanel.add(new JButton("Center"), BorderLayout.CENTER);

      JPanel cardPanel = new JPanel(new CardLayout());
      cardPanel.add(new JButton("Card 1"), "Card 1");
      cardPanel.add(new JButton("Card 2"), "Card 2");
      cardPanel.add(new JButton("Card 3"), "Card 3");

      JButton switchCardButton = new JButton("Switch Card");
      switchCardButton.addActionListener(new ActionListener() {
         private int cardIndex = 0;
         @Override
         public void actionPerformed(ActionEvent e) {
```

```
            cardIndex = (cardIndex + 1) % 3;
            CardLayout cl = (CardLayout) (cardPanel.getLayout());
            cl.show(cardPanel, "Card " + (cardIndex + 1));
        }
    });

    frame.add(flowPanel, BorderLayout.NORTH);
    frame.add(gridPanel, BorderLayout.WEST);
    frame.add(borderPanel, BorderLayout.CENTER);
    frame.add(cardPanel, BorderLayout.EAST);
    frame.add(switchCardButton, BorderLayout.SOUTH);

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

Output :

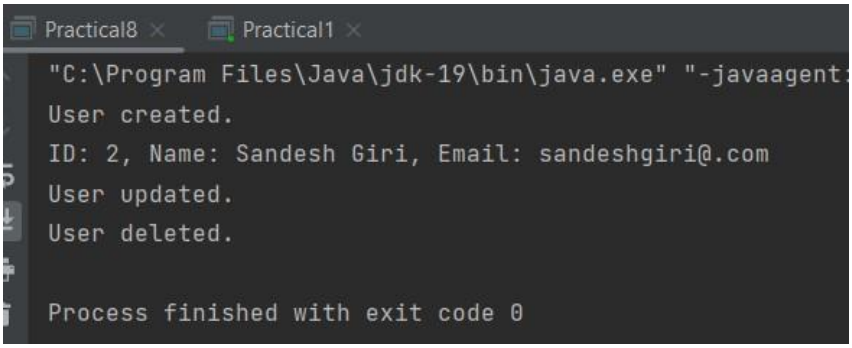


8. Show a step to connect java program with database with CRUD operation.

```
import java.sql.*;
public class Practical8 {
   private static final String URL = "jdbc:mysql://localhost:3306/testdb";
   private static final String USER = "root";
   private static final String PASSWORD = "";

   public static void main(String[] args) {
      try (Connection connection = DriverManager.getConnection(URL, USER,
PASSWORD)) {
         createUser(connection, "Sandesh Giri", "sandeshgiri@.com");
         readUsers(connection);
         updateUser(connection, 1, "Sandesh Giri", "sandeshgiri@gmail.com");
         deleteUser(connection, 1);
      } catch (SQLException e) {
         e.printStackTrace();
      }
   }
   public static void createUser(Connection connection, String name, String email)
throws SQLException {
      String sql = "INSERT INTO usersone (name, email) VALUES (?, ?)";
      try (PreparedStatement statement = connection.prepareStatement(sql)) {
         statement.setString(1, name);
         statement.setString(2, email);
         statement.executeUpdate();
         System.out.println("User created.");
      }
   }
   public static void readUsers(Connection connection) throws SQLException {
      String sql = "SELECT * FROM usersone";
      try (Statement statement = connection.createStatement();
          ResultSet resultSet = statement.executeQuery(sql)) {

         while (resultSet.next()) {
            int id = resultSet.getInt("id");
            String name = resultSet.getString("name");
            String email = resultSet.getString("email");
            System.out.println("ID: " + id + ", Name: " + name + ", Email: " + email); }
      }
   }
   public static void updateUser(Connection connection, int id, String newName, String
newEmail) throws SQLException {
```

```
        String sql = "UPDATE usersone SET name = ?, email = ? WHERE id = ?";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setString(1, newName);
            statement.setString(2, newEmail);
            statement.setInt(3, id);
            statement.executeUpdate();
            System.out.println("User updated.");
        }
    }
    public static void deleteUser(Connection connection, int id) throws SQLException {
        String sql = "DELETE FROM usersone WHERE id = ?";
        try (PreparedStatement statement = connection.prepareStatement(sql)) {
            statement.setInt(1, id);
            statement.executeUpdate();
            System.out.println("User deleted."); }
    }
}
```

Output :



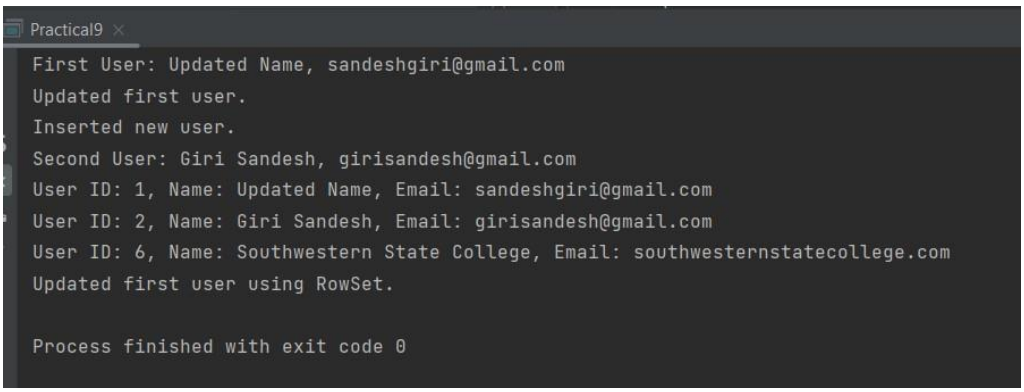
9. Demonstrate the use of row set and updatable and scrollable result set.

```
import java.sql.*;
import javax.sql.rowset.RowSetProvider;
import javax.sql.rowset.CachedRowSet;

public class Practical9 {
   private static final String URL = "jdbc:mysql://localhost:3306/testdb";
   private static final String USER = "root";
   private static final String PASSWORD = "";

   public static void main(String[] args) {
      try (Connection connection = DriverManager.getConnection(URL, USER,
PASSWORD)) {
         scrollableAndUpdatableResultSet(connection);

         rowSetExample();
      } catch (SQLException e) {
         e.printStackTrace();
      }
   }
   public static void scrollableAndUpdatableResultSet(Connection connection) throws
SQLException {
      String sql = "SELECT * FROM userstwo";

      Statement statement = connection.createStatement(
            ResultSet.TYPE_SCROLL_INSENSITIVE, // Scrollable ResultSet
            ResultSet.CONCUR_UPDATABLE         // Updatable ResultSet
      );

      ResultSet resultSet = statement.executeQuery(sql);

      if (!resultSet.isBeforeFirst()) {
         System.out.println("No data found in userstwo table.");
         return;
      }

      if (resultSet.last()) {
         System.out.println("Last User: " + resultSet.getString("name") + ", " +
resultSet.getString("email"));
      }

      resultSet.first();
```

```
        System.out.println("First User: " + resultSet.getString("name") + ", " +
resultSet.getString("email"));

        resultSet.updateString("name", "Updated Name");
        resultSet.updateRow();
        System.out.println("Updated first user.");

        resultSet.moveToInsertRow();
        resultSet.updateString("name", "Southwestern State College");
        resultSet.updateString("email", "southwesternstatecollege.com");
        resultSet.insertRow();
        System.out.println("Inserted new user.");

        resultSet.absolute(2);
        System.out.println("Second User: " + resultSet.getString("name") + ", " +
resultSet.getString("email"));
        resultSet.close();
        statement.close();
    }

    // Method to demonstrate RowSet
    public static void rowSetExample() throws SQLException {
        CachedRowSet rowSet = RowSetProvider.newFactory().createCachedRowSet();
        rowSet.setUrl(URL);
        rowSet.setUsername(USER);
        rowSet.setPassword(PASSWORD);

        rowSet.setCommand("SELECT * FROM userstwo");

        rowSet.execute();

        if (!rowSet.isBeforeFirst()) {
            System.out.println("No data found in userstwo table.");
            return;
        }

        while (rowSet.next()) {
            System.out.println("User ID: " + rowSet.getInt("id") + ", Name: " +
rowSet.getString("name") + ", Email: " + rowSet.getString("email"));
        }

        if (rowSet.first()) {
            rowSet.updateString("name", "RowSet Updated User");
```

```
            rowSet.updateRow();
            System.out.println("Updated first user using RowSet.");
        }

        rowSet.close();
    }
}
```

Output :



10. Create one complete form using swing (id,username, password, repassword, gender (radio button), course (check box), country (combo box), submit and reset button). validate the form (check for emptiness, password and repassword should be equal) and insert the data into database using prepared statement. After this create a login form with fields username, password and submit button. if username and password matched with database record then display all the record of table using rowset or scrollable ad updateable result set. Also perform following operation:

- Update the name of student to “sanu” and course to “BCA” whose id is 3
- Update country to “USA” for the student whose id is 5.
- Delete the record of employee whose name is “sam”.

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

public class Practical10 {
   public static void main(String[] args) {
      JFrame frame = new JFrame("Registration and Login");
      frame.setSize(400, 300);
      frame.setLayout(null);
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      JButton registrationButton = new JButton("Register");
      registrationButton.setBounds(50, 50, 150, 30);
      JButton loginButton = new JButton("Login");
      loginButton.setBounds(220, 50, 150, 30);
      frame.add(registrationButton);
      frame.add(loginButton);

      registrationButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            new RegistrationForm();
         }
      });
      loginButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            new LoginForm();
         }
      });

      frame.setVisible(true);
```

```
    }
}
class RegistrationForm {
    private JFrame frame;
    private JTextField idField, usernameField, passwordField, repasswordField;
    private JRadioButton maleButton, femaleButton;
    private JCheckBox bcaCheckBox, mcaCheckBox;
    private JComboBox<String> countryComboBox;
    public RegistrationForm() {
        frame = new JFrame("Registration Form");
        frame.setSize(400, 400);
        frame.setLayout(null);
        frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

        JLabel idLabel = new JLabel("ID:");
        idLabel.setBounds(30, 20, 100, 30);
        idField = new JTextField();
        idField.setBounds(150, 20, 200, 30);

        JLabel usernameLabel = new JLabel("Username:");
        usernameLabel.setBounds(30, 60, 100, 30);
        usernameField = new JTextField();
        usernameField.setBounds(150, 60, 200, 30);

        JLabel passwordLabel = new JLabel("Password:");
        passwordLabel.setBounds(30, 100, 100, 30);
        passwordField = new JPasswordField();
        passwordField.setBounds(150, 100, 200, 30);

        JLabel repasswordLabel = new JLabel("Re-enter   Password:");
        repasswordLabel.setBounds(30, 140, 150, 30);
        repasswordField = new JPasswordField();
        repasswordField.setBounds(150, 140, 200, 30);

        JLabel genderLabel = new JLabel("Gender:");
        genderLabel.setBounds(30, 180, 100, 30);
        maleButton = new JRadioButton("Male");
        maleButton.setBounds(150, 180, 80, 30);
        femaleButton = new JRadioButton("Female");
        femaleButton.setBounds(230, 180, 80, 30);
        ButtonGroup genderGroup = new ButtonGroup();
        genderGroup.add(maleButton);
        genderGroup.add(femaleButton);
```

```
        JLabel courseLabel = new JLabel("Course:");
        courseLabel.setBounds(30, 220, 100, 30);
        bcaCheckBox = new JCheckBox("BCA");
        bcaCheckBox.setBounds(150, 220, 80, 30);
        mcaCheckBox = new JCheckBox("MCA");
        mcaCheckBox.setBounds(230, 220, 80, 30);

        JLabel countryLabel = new JLabel("Country:");
        countryLabel.setBounds(30, 260, 100, 30);
        countryComboBox = new JComboBox<>(new String[] {"Select", "USA",
"Canada", "India"});
        countryComboBox.setBounds(150, 260, 200, 30);

        JButton submitButton = new JButton("Submit");
        submitButton.setBounds(100, 300, 100, 30);
        JButton resetButton = new JButton("Reset");
        resetButton.setBounds(220, 300, 100, 30);

        frame.add(idLabel);
        frame.add(idField);
        frame.add(usernameLabel);
        frame.add(usernameField);
        frame.add(passwordLabel);
        frame.add(passwordField);
        frame.add(repasswordLabel);
        frame.add(repasswordField);
        frame.add(genderLabel);
        frame.add(maleButton);
        frame.add(femaleButton);
        frame.add(courseLabel);
        frame.add(bcaCheckBox);
        frame.add(mcaCheckBox);
        frame.add(countryLabel);
        frame.add(countryComboBox);
        frame.add(submitButton);
        frame.add(resetButton);

        submitButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String id = idField.getText();
                String username = usernameField.getText();
```

```
            String password = passwordField.getText();
            String repassword = repasswordField.getText();
            String gender = maleButton.isSelected() ? "Male" : "Female";
            String course = (bcaCheckBox.isSelected() ? "BCA " : "") +
(mcaCheckBox.isSelected() ? "MCA" : "");
            String country = (String) countryComboBox.getSelectedItem();

            if (id.isEmpty() || username.isEmpty() || password.isEmpty() ||
repassword.isEmpty() || country.equals("Select")) {
               JOptionPane.showMessageDialog(frame, "All fields must be filled!");
            } else if (!password.equals(repassword)) {
               JOptionPane.showMessageDialog(frame, "Passwords do not match!");
            } else {
               try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb", "root", "")) {
                  String sql = "INSERT INTO db_ten (id, username, password, gender,
course, country) VALUES (?, ?, ?, ?, ?, ?)";
                  try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
                     pstmt.setInt(1, Integer.parseInt(id));
                     pstmt.setString(2, username);
                     pstmt.setString(3, password);
                     pstmt.setString(4, gender);
                     pstmt.setString(5, course);
                     pstmt.setString(6, country);
                     pstmt.executeUpdate();
                     JOptionPane.showMessageDialog(frame, "Registration Successful!");
                  }
               } catch (SQLException ex) {
                  ex.printStackTrace();
                  JOptionPane.showMessageDialog(frame, "Database error: " +
ex.getMessage());
               }
            }
         }
      });

      resetButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            idField.setText("");
            usernameField.setText("");
            passwordField.setText("");
            repasswordField.setText("");
```

```
            genderGroup.clearSelection();
            bcaCheckBox.setSelected(false);
            mcaCheckBox.setSelected(false);
            countryComboBox.setSelectedIndex(0);
        }
    });

    frame.setVisible(true);
  }
}
class LoginForm {
  private JFrame frame;
  private JTextField usernameField, passwordField;
  public LoginForm() {
    frame = new JFrame("Login Form");
    frame.setSize(400, 200);
    frame.setLayout(null);
    frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

    JLabel usernameLabel = new JLabel("Username:");
    usernameLabel.setBounds(30, 20, 100, 30);
    usernameField = new JTextField();
    usernameField.setBounds(150, 20, 200, 30);

    JLabel passwordLabel = new JLabel("Password:");
    passwordLabel.setBounds(30, 60, 100, 30);
    passwordField = new JPasswordField();
    passwordField.setBounds(150, 60, 200, 30);

    JButton submitButton = new JButton("Submit");
    submitButton.setBounds(150, 100, 100, 30);

    frame.add(usernameLabel);
    frame.add(usernameField);
    frame.add(passwordLabel);
    frame.add(passwordField);
    frame.add(submitButton);
    submitButton.addActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        String username = usernameField.getText();
        String password = passwordField.getText();
        try (Connection conn =
```

```
DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb", "root", "")) {
                String sql = "SELECT * FROM db_ten WHERE username = ? AND
password = ?";
                try (PreparedStatement pstmt = conn.prepareStatement(sql)) {
                   pstmt.setString(1, username);
                   pstmt.setString(2, password);
                   try (ResultSet rs = pstmt.executeQuery()) {
                      if (rs.next()) {
                         new UpdateDeleteForm();
                         frame.dispose();
                      } else {
                         JOptionPane.showMessageDialog(frame, "Invalid username or
password!");
                      }
                   }
                }
             } catch (SQLException ex) {
                ex.printStackTrace();
                JOptionPane.showMessageDialog(frame, "Database error: " +
ex.getMessage());
             }
          }
       });
       frame.setVisible(true);
   }
}
class UpdateDeleteForm {
   private JFrame frame;
   private JTextArea textArea;
   public UpdateDeleteForm() {
      frame = new JFrame("Update/Delete Records");
      frame.setSize(600, 400);
      frame.setLayout(null);
      frame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);

      textArea = new JTextArea();
      textArea.setBounds(30, 20, 540, 250);
      textArea.setEditable(false);
      JScrollPane scrollPane = new JScrollPane(textArea);
      scrollPane.setBounds(30, 20, 540, 250);
      frame.add(scrollPane);

      JButton updateButton = new JButton("Update");
```

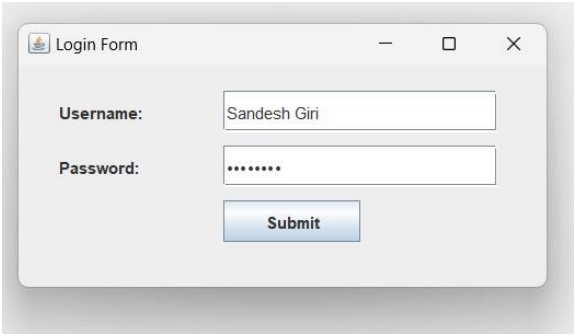
```
        updateButton.setBounds(100, 300, 100, 30);
        JButton deleteButton = new JButton("Delete");
        deleteButton.setBounds(220, 300, 100, 30);
        JButton refreshButton = new JButton("Refresh");
        refreshButton.setBounds(340, 300, 100, 30);

        frame.add(updateButton);
        frame.add(deleteButton);
        frame.add(refreshButton);

        refreshButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                loadRecords();
            }
        });

        updateButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb", "root", "")) {
                    String sql = "UPDATE db_ten SET username = 'sanu', course = 'BCA'
WHERE id = 3";
                    try (Statement stmt = conn.createStatement()) {
                        stmt.executeUpdate(sql);
                        JOptionPane.showMessageDialog(frame, "Record updated!");
                        loadRecords();
                    }
                } catch (SQLException ex) {
                    ex.printStackTrace();
                    JOptionPane.showMessageDialog(frame, "Database error: " +
ex.getMessage());
                }
            }
        });

        deleteButton.addActionListener(new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb", "root", "")) {
                    String sql = "DELETE FROM db_ten WHERE username = 'sam'";
```

```
                try (Statement stmt = conn.createStatement()) {
                    stmt.executeUpdate(sql);
                    JOptionPane.showMessageDialog(frame, "Record deleted!");
                    loadRecords();
                }
            } catch (SQLException ex) {
                ex.printStackTrace();
                JOptionPane.showMessageDialog(frame, "Database error: " +
ex.getMessage());
            }
        }
    });

    loadRecords();

    frame.setVisible(true);
  }

  private void loadRecords() {
    textArea.setText("");
    try (Connection conn =
DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb", "root", "")) {
        String sql = "SELECT * FROM db_ten";
        try (Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery(sql)) {
            while (rs.next()) {
                textArea.append("ID: " + rs.getInt("id") +
                        ", Username: " + rs.getString("username") +
                        ", Password: " + rs.getString("password") +
                        ", Gender: " + rs.getString("gender") +
                        ", Course: " + rs.getString("course") +
                        ", Country: " + rs.getString("country") + "\n");
            }
        }
    } catch (SQLException ex) {
        ex.printStackTrace();
        JOptionPane.showMessageDialog(frame, "Database error: " + ex.getMessage());
    }
  }
}
```

Register or Login

Registration Form

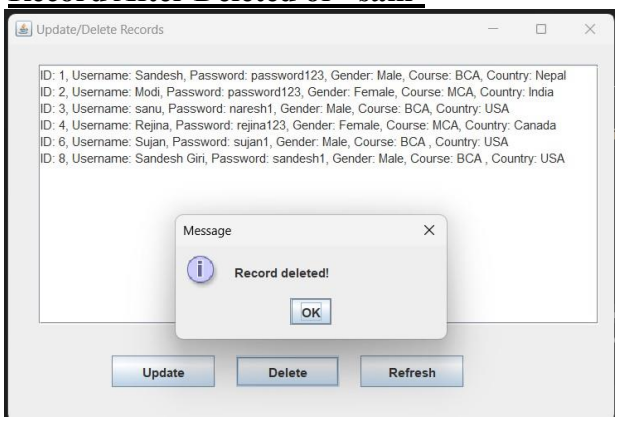
Login Form



Record Before Update

Records After Updated to sanu

Record After Deleted of ‘sam’

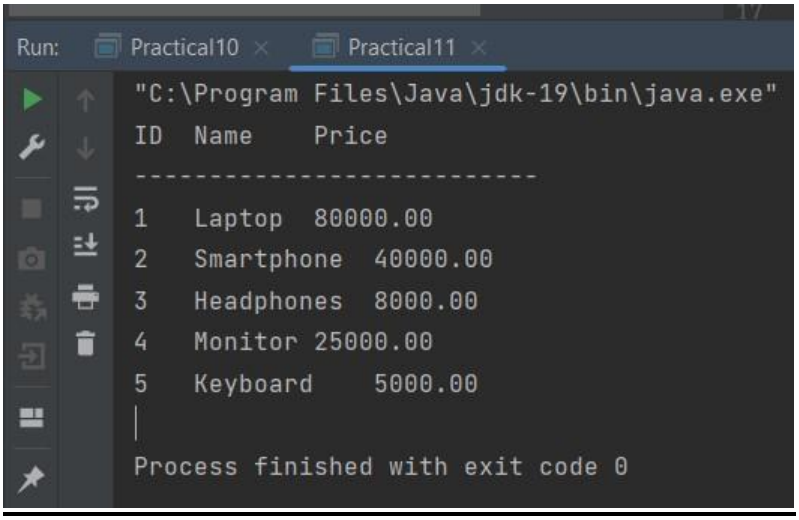


11. Write a program to demonstrate the use of result set to retrieve data from database.

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class Practical11 {
   private static final String DB_URL = "jdbc:mysql://localhost:3306/testdb";
   private static final String USER = "root";
   private static final String PASS = "";

   public static void main(String[] args) {
      Connection conn = null;
      Statement stmt = null;
      ResultSet rs = null;

      try {
         Class.forName("com.mysql.cj.jdbc.Driver");
         conn = DriverManager.getConnection(DB_URL, USER, PASS);
         stmt = conn.createStatement();
         String sql = "SELECT id, name, price FROM products";
         rs = stmt.executeQuery(sql);

         System.out.println("ID\tName\tPrice");
         System.out.println("--------------------------");
         while (rs.next()) {
            // Retrieve by column name
            int id = rs.getInt("id");
            String name = rs.getString("name");
            double price = rs.getDouble("price");

            System.out.printf("%d\t%s\t%.2f\n", id, name, price);
         }
      } catch (SQLException se) {
         se.printStackTrace();
      } catch (ClassNotFoundException e) {
         e.printStackTrace();
      } finally {
         try {
            if (rs != null) rs.close();
            if (stmt != null) stmt.close();
```

```
            if (conn != null) conn.close();
        } catch (SQLException se) {
            se.printStackTrace();
        }
    }
  }
}
```

Output :



12. Write a full program to demonstrate different ways that can be used to run servlet program?

HelloServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
@WebServlet("/hello")
public class HelloServlet extends HttpServlet {

   @Override
   protected void doGet(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/html");
      PrintWriter out = response.getWriter();
      out.println("<html><body>");
      out.println("<h2>Hello, this is a servlet response!</h2>");
      out.println("</body></html>");
   }
}
```

web.xml

```
<web-app xmlns="http://xmlns.jcp.org/xml/ns/javaee"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="http://xmlns.jcp.org/xml/ns/javaee
http://xmlns.jcp.org/xml/ns/javaee/web-app_4_0.xsd"
       version="4.0">

   <servlet>
      <servlet-name>HelloServlet</servlet-name>
      <servlet-class>HelloServlet</servlet-class>
   </servlet>

   <servlet-mapping>
      <servlet-name>HelloServlet</servlet-name>
      <url-pattern>/hello</url-pattern>
   </servlet-mapping>
</web-app>
```

13. Write a program to demonstrate how form property can be accessed using servlet?

Step 1: Creating an HTML form
form.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Form Submission</title>
</head>
<body>
   <h2>Enter Your Details</h2>
   <form action="FormServlet" method="post">
      Name: <input type="text" name="name"><br><br>
      Email: <input type="text" name="email"><br><br>
      Age: <input type="text" name="age"><br><br>
      <input type="submit" value="Submit">
   </form>
</body>
</html>
```

Step 2: Create the Servlet
FormServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/FormServlet")
public class FormServlet extends HttpServlet {

   @Override
   protected void doPost(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/html");

      String name = request.getParameter("name");
```

```
        String email = request.getParameter("email");
        String age = request.getParameter("age");

        PrintWriter out = response.getWriter();
        out.println("<html><body>");
        out.println("<h2>Form Data Received</h2>");
        out.println("<p>Name: " + name + "</p>");
        out.println("<p>Email: " + email + "</p>");
        out.println("<p>Age: " + age + "</p>");
        out.println("</body></html>");
    }
}
```

14. Write a program to show setting and retrieving cookie in servlet?

Step 1: Create
SetCookieServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/SetCookieServlet")
public class SetCookieServlet extends HttpServlet {

   @Override
   protected void doGet(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/html");

      Cookie cookie = new Cookie("username", "JohnDoe");
      cookie.setMaxAge(24 * 60 * 60);

      response.addCookie(cookie);
      PrintWriter out = response.getWriter();
      out.println("<html><body>");
      out.println("<h2>Cookie has been set in your browser.</h2>");
      out.println("<p><a href='GetCookieServlet'>Retrieve Cookie</a></p>");
      out.println("</body></html>");
   }
}
```

Step 2: Create
GetCookieServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
```

```
import javax.servlet.http.HttpServletResponse;

@WebServlet("/GetCookieServlet")
public class GetCookieServlet extends HttpServlet {

   @Override
   protected void doGet(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/html");

      Cookie[] cookies = request.getCookies();

      PrintWriter out = response.getWriter();
      out.println("<html><body>");

      if (cookies != null) {
         out.println("<h2>Retrieved Cookies:</h2>");
         boolean found = false;

         for (Cookie cookie : cookies) {
            if ("username".equals(cookie.getName())) {
               out.println("<p>Username: " + cookie.getValue() + "</p>");
               found = true;
               break;
            }
         }

         if (!found) {
            out.println("<p>No 'username' cookie found.</p>");
         }
      } else {
         out.println("<p>No cookies found.</p>");
      }

      out.println("</body></html>");
   }
}
```

15. Make a web form of your choice then validate it using servlet?

Step 1: Creating HTML form
form.html

```
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="UTF-8">
   <title>Form Validation</title>
</head>
<body>
   <h2>User Registration</h2>
   <form action="ValidateFormServlet" method="post">
      Name: <input type="text" name="name"><br><br>
      Email: <input type="text" name="email"><br><br>
      Age: <input type="text" name="age"><br><br>
      <input type="submit" value="Submit">
   </form>
</body>
</html>
```

Step 2: Create Servlet
ValidateFormServlet.java

```
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/ValidateFormServlet")
public class ValidateFormServlet extends HttpServlet {

   @Override
   protected void doPost(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      response.setContentType("text/html");
      String name = request.getParameter("name");
      String email = request.getParameter("email");
      String ageStr = request.getParameter("age");
```

```
        PrintWriter out = response.getWriter();
        out.println("<html><body>");

        if (name.isEmpty()) {
            out.println("<p style='color:red;'>Name is required.</p>");
        } else if (email.isEmpty() || !email.contains("@") || !email.contains(".")) {
            out.println("<p style='color:red;'>Invalid email.</p>");
        } else {
            try {
                int age = Integer.parseInt(ageStr);
                if (age <= 0) {
                    out.println("<p style='color:red;'>Age must be greater than 0.</p>");
                } else {
                    out.println("<h2>Registration Successful</h2>");
                    out.println("<p>Name: " + name + "</p>");
                    out.println("<p>Email: " + email + "</p>");
                    out.println("<p>Age: " + age + "</p>");
                }
            } catch (NumberFormatException e) {
                out.println("<p style='color:red;'>Invalid age.</p>");
            }
        }
        out.println("</body></html>");
    }
}
```

16. Write a program to show how jsp can be executed?

Step 1: Creating the JSP file
hello.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<!DOCTYPE html>
<html lang="en">
<head>
   <meta charset="ISO-8859-1">
   <title>JSP Execution Example</title>
</head>
<body>
   <h2>Welcome to JSP Execution Example!</h2>
   <p>Hello, <%= request.getAttribute("username") %>! This message was passed from
the servlet.</p>
</body>
</html>
```

Step 2: Create the Servlet
HelloServlet.java

```
import java.io.IOException;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

@WebServlet("/HelloServlet")
public class HelloServlet extends HttpServlet {

   @Override
   protected void doGet(HttpServletRequest request, HttpServletResponse response)
         throws ServletException, IOException {
      request.setAttribute("username", "Sandesh Giri");

      request.getRequestDispatcher("hello.jsp").forward(request, response);
   }
}
```

17. Write a program to demonstrate different tags of jsp?

Step 1: Create the JSP file
tagsDemo.jsp

```
<%@ page language="java" contentType="text/html; charset=ISO-8859-1"
pageEncoding="ISO-8859-1"%>
<%@ page import="java.util.Date" %>
<!DOCTYPE html> <html lang="en">
<head>
   <meta charset="ISO-8859-1">
   <title>JSP Tag Demonstration</title>
</head>
<body>
   <h2>Demonstrating Different JSP Tags</h2>
   <p>Content Type: <%= response.getContentType() %></p>
   <%
      String greeting = "Hello, JSP!";
      out.println("<p>Greeting from Scriptlet: " + greeting + "</p>");
   %>
   <p>Current Date and Time: <%= new Date() %></p>
   <%!
      public String getWelcomeMessage(String name) {
         return "Welcome, " + name + "!";  }  %>
   <p>Welcome Message using Declaration Tag: <%= getWelcomeMessage("John")
%></p>
   <jsp:useBean id="user" class="User" scope="session"/>
   <jsp:setProperty name="user" property="name" value="Alice"/>
   <p>User Name from JavaBean: <jsp:getProperty name="user"
property="name"/></p>
</body></html>
```

Step 2: Create JavaBean Class
User.java

```
public class User {
   private String name;
   public String getName() {
      return name;
   }
   public void setName(String name) {
      this.name = name;
   }}
```

18. Write a program to demonstrate client server application using RMI.

Step 1: Create the Remote Interface
Hello.java

```
import java.rmi.Remote;
import java.rmi.RemoteException;

public interface Hello extends Remote {
   String sayHello(String name) throws RemoteException;
}
```

Step 2: Create the Server Implementation (HelloImpl.java)

```
import java.rmi.RemoteException;
import java.rmi.server.UnicastRemoteObject;

public class HelloImpl extends UnicastRemoteObject implements Hello {

   public HelloImpl() throws RemoteException {
      super();
   }

   @Override
   public String sayHello(String name) throws RemoteException {
      return "Hello, " + name + "!";
   }
}
```

Step 3: Create the Server
RMIServer.java

```
import java.rmi.Naming;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;

public class RMIServer {

   public static void main(String[] args) {
      try {
```

```
            Registry registry = LocateRegistry.createRegistry(1099); // Default RMI port

            HelloImpl hello = new HelloImpl();

            Naming.rebind("rmi://localhost/Hello", hello);

            System.out.println("RMI Server is ready.");
        } catch (Exception e) {
            System.out.println("Server exception: " + e.getMessage());
            e.printStackTrace();
        }
    }
}
```

Step 4: Create the Client

RMIClient.java

```
import java.rmi.Naming;
public class RMIClient {
    public static void main(String[] args) {
        try {
            Hello hello = (Hello) Naming.lookup("rmi://localhost/Hello");

            String response = hello.sayHello("Sandesh");

            System.out.println(response);
        } catch (Exception e) {
            System.out.println("Client exception: " + e.getMessage());
            e.printStackTrace();
        }
    }
}
```