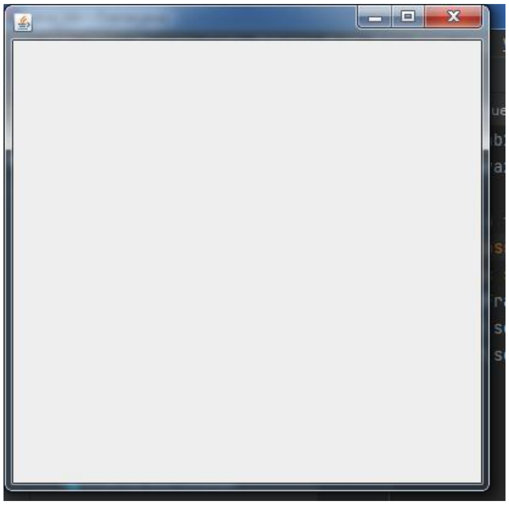
## LAB 1

1. Create a frame with size 400X400 using swing.

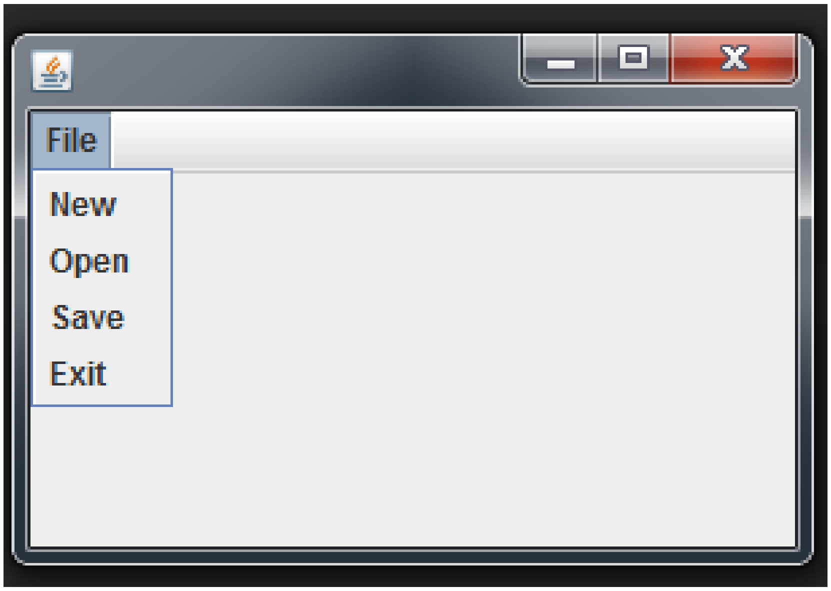
```
package lab1.ques1;
import javax.swing.*;
//1. Create a frame with size 400X400 using swing.
public class Frame {
  public static void main(String[] args) {
    JFrame f = new JFrame();//creating instance of JFrame
    f.setSize(400, 400);//400 width and 500 height
    f.setVisible(true);
  }
}
```



2. Demonstrate menu implementation frame using swing.

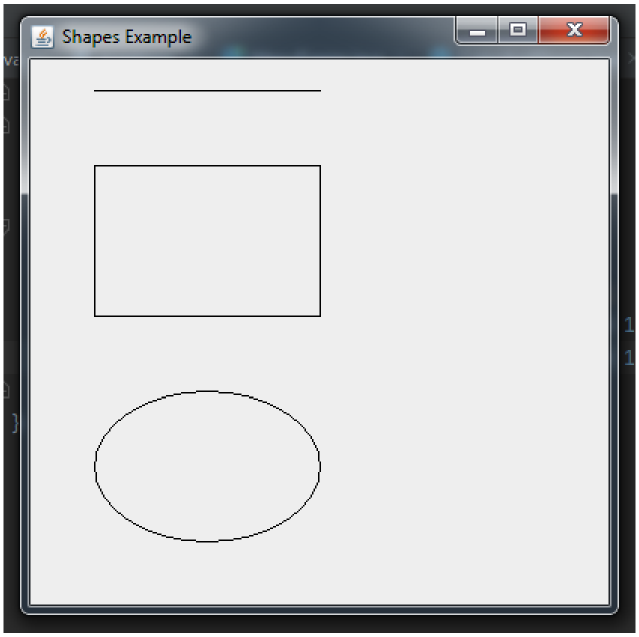
```
package lab1.ques2;
import javax.swing.*;
import java.awt.event.*;
//Demonstrate menu implementation frame using swing
public class MenuFrame extends JFrame {
  private JMenuBar menuBar;
  private JMenu fileMenu;
  private JMenuItem newItem;
  private JMenuItem openItem;
  private JMenuItem saveItem;
  private JMenuItem exitItem;
  public MenuFrame() {
    menuBar = new JMenuBar();
    fileMenu = new JMenu("File");
    newItem = new JMenuItem("New");
    openItem = new JMenuItem("Open");
    saveItem = new JMenuItem("Save");
```

```
      exitItem = new JMenuItem("Exit");
      fileMenu.add(newItem);
      fileMenu.add(openItem);
      fileMenu.add(saveItem);
      fileMenu.add(exitItem);
      menuBar.add(fileMenu);
      setJMenuBar(menuBar);
      exitItem.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
            System.exit(0);
         }
      });
      setSize(300, 200);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setVisible(true);
   }
   public static void main(String[] args) {
      SwingUtilities.invokeLater(new Runnable() {
         public void run() {
            new MenuFrame();
         }
      });
   }
}
```



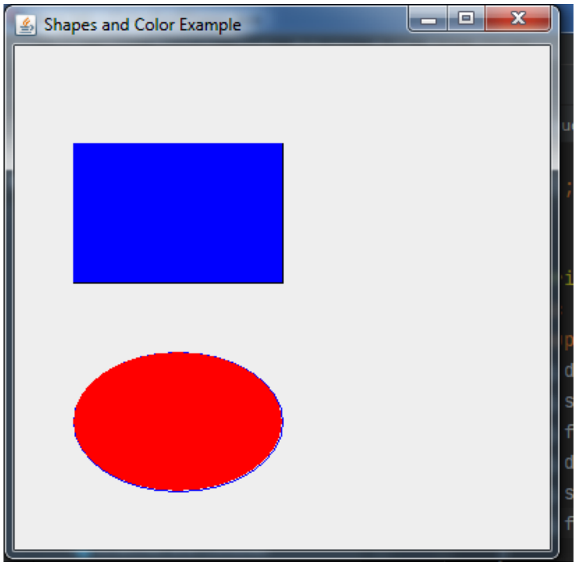
3. Draw line, rectangle, circle using graphics in swing.

```
package lab1;
import javax.swing.*;
import java.awt.*;
//3. Draw line, rectangle, circle using graphics in swing.
public class Shapes extends JFrame {
    public Shapes() {
        setTitle("Shapes Example");
        setSize(400, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new Shapes();
            }
        });
    }
    @Override
    public void paint(Graphics g) {
        super.paint(g);
        g.drawLine(50, 50, 200, 50);
        g.drawRect(50, 100, 150, 100);
        g.drawOval(50, 250, 150, 100);
    }
}
```



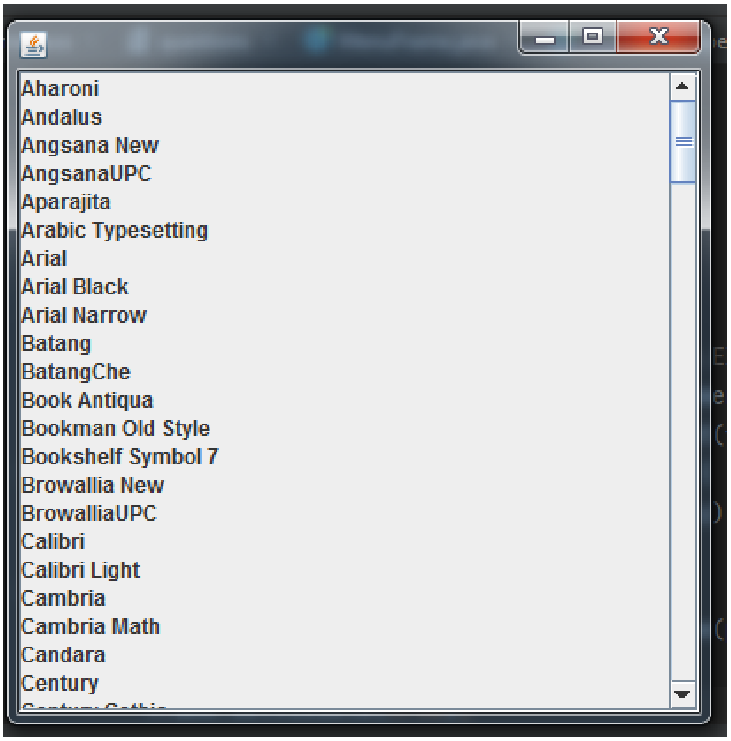
4. Demonstrate color and shape with color filled inside using swing.

```
package lab1;
import javax.swing.*;
import java.awt.*;
//4. Demonstrate color and shape with color filled inside using swing.
public class ColorAndShape extends JFrame {
   public ColorAndShape() {
      setTitle("Shapes and Color Example");
      setSize(400, 400);
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setVisible(true);
   }
   public static void main(String[] args) {
      SwingUtilities.invokeLater(new Runnable() {
         public void run() {
            new ColorAndShape();
         }
      });
   }
   @Override
   public void paint(Graphics g) {
      super.paint(g);
      g.drawRect(50, 100, 150, 100);
      g.setColor(Color.BLUE);
      g.fillRect(50, 100, 150, 100);
      g.drawOval(50, 250, 150, 100);
      g.setColor(Color.RED);
      g.fillOval(50, 250, 150, 100);
   }
}
```



5. Write a program to list all the available font family.

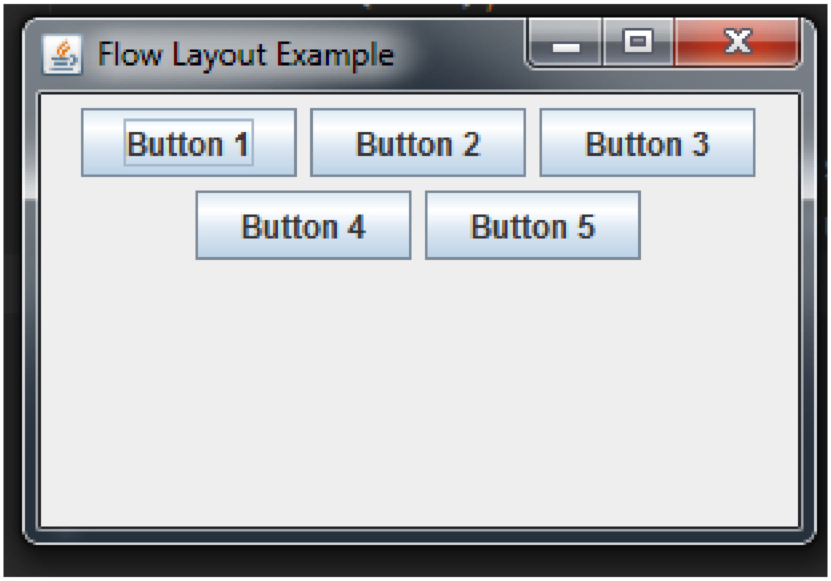
```
package lab1.ques5;
import javax.swing.*;
import java.awt.*;
public class Fonts {
  public static void main(String[] args) {
    JFrame frame = new JFrame();
    frame.setSize(400, 400);
    //frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    String[] fontFamilies =
GraphicsEnvironment.getLocalGraphicsEnvironment().getAvailableFontFamilyNames();
    JPanel panel = new JPanel(new GridLayout(fontFamilies.length, 1));
    for (String fontFamily : fontFamilies) {
      JLabel label = new JLabel(fontFamily);
      panel.add(label);
    }
    JScrollPane scrollPane = new JScrollPane(panel);
    frame.add(scrollPane);
    frame.setVisible(true);
  }
}
```



```
Aharoni
Andalus
Angsana New
AngsanaUPC
Aparajita
Arabic Typesetting
Arial
Arial Black
Arial Narrow
Batang
BatangChe
Book Antiqua
Bookman Old Style
Bookshelf Symbol 7
Browallia New
BrowalliaUPC
Calibri
Calibri Light
Cambria
Cambria Math
Candara
Century
```

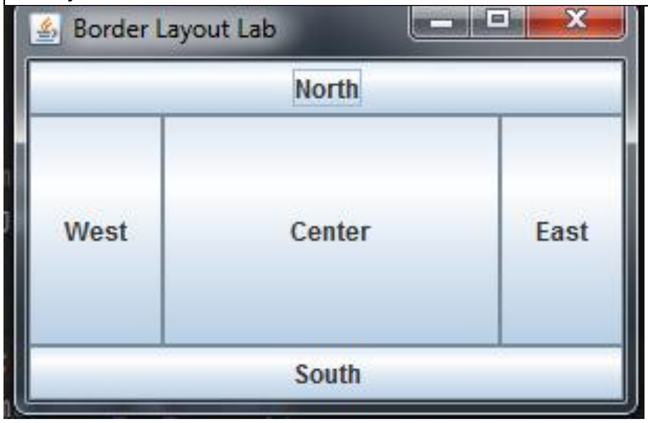
6. Write a program to show implementation of flow layout.

```
package lab1.ques6;
import javax.swing.*;
import java.awt.*;
//6. Write a program to show implementation of flow layout.
public class FlowLayoutLab extends JFrame {
    public FlowLayoutLab() {
        setTitle("Flow Layout Lab");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // Create a JPanel with FlowLayout
        JPanel panel = new JPanel(new FlowLayout());
        // Create buttons and add them to the panel
        for (int i = 1; i <= 5; i++) {
            JButton button = new JButton("Button " + i);
            panel.add(button);
        }
        // Add the panel to the frame
        add(panel);
        setSize(300, 200);
        setVisible(true);
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new FlowLayoutLab();
            }
        });
    }
}
```



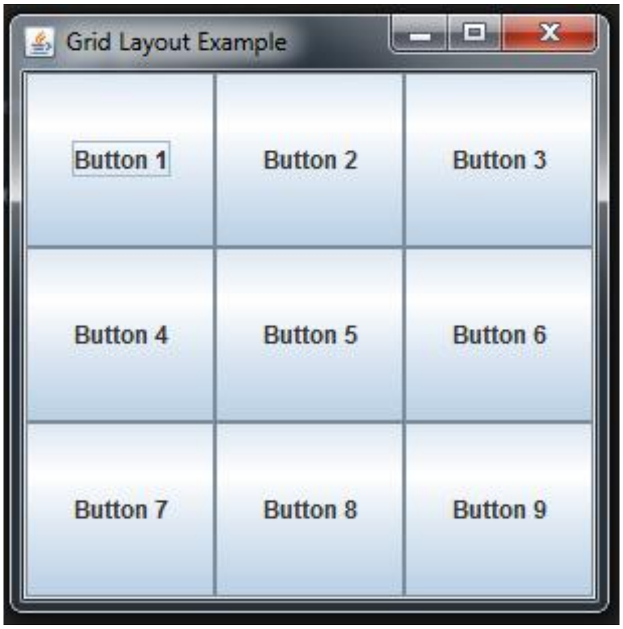
7. Write a program to show implementation of border layout.

```
package lab1.ques7;
import javax.swing.*;
import java.awt.*;
//7.Write a program to show implementation of border layout.
public class BorderLayoutLab extends JFrame {
   public BorderLayoutLab() {
      setTitle("Border Layout Lab");
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      // Create a JPanel with BorderLayout
      JPanel panel = new JPanel(new BorderLayout());
      // Create buttons and add them to the panel
      JButton northButton = new JButton("North");
      JButton southButton = new JButton("South");
      JButton eastButton = new JButton("East");
      JButton westButton = new JButton("West");
      JButton centerButton = new JButton("Center");
      panel.add(northButton, BorderLayout.NORTH);
      panel.add(southButton, BorderLayout.SOUTH);
      panel.add(eastButton, BorderLayout.EAST);
      panel.add(westButton, BorderLayout.WEST);
      panel.add(centerButton, BorderLayout.CENTER);
      // Add the panel to the frame
      add(panel);
      setSize(300, 200);
      setVisible(true);
   }
   public static void main(String[] args) {
      SwingUtilities.invokeLater(new Runnable() {
         public void run() {
            new BorderLayoutLab();
         }
      });
   }
}
```



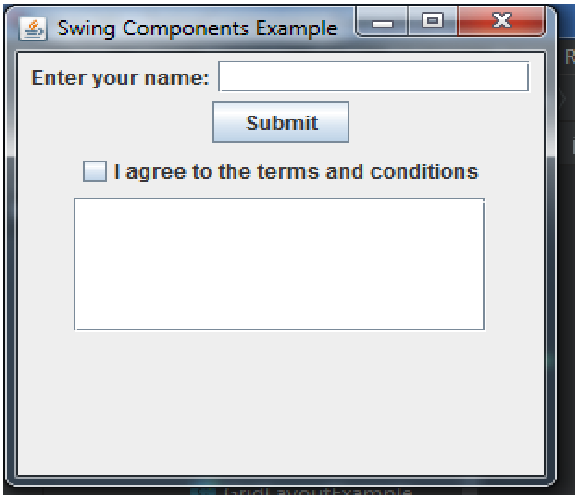
8. Write a program to show implementation of grid layout.

```
package lab1.ques8;
import javax.swing.*;
import java.awt.*;
public class GridLayoutExample extends JFrame {
  public GridLayoutExample() {
    setTitle("Grid Layout Example");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    // Create a JPanel with GridLayout
    JPanel panel = new JPanel(new GridLayout(3, 3));
    // Create buttons and add them to the panel
    for (int i = 1; i <= 9; i++) {
      JButton button = new JButton("Button " + i);
      panel.add(button);
    }
    // Add the panel to the frame
    add(panel);
    setSize(300, 300);
    setVisible(true);
  }
  public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        new GridLayoutExample();
      }
    });
  }
}
```



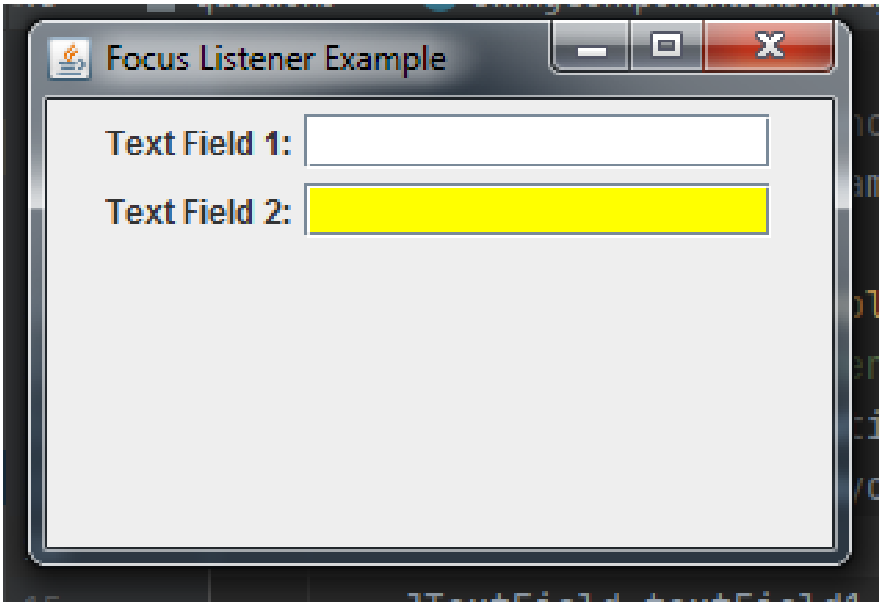
9. Write a program to use any five components in swing.

```
package lab1.ques9;
import javax.swing.*;
import java.awt.*;
public class SwingComponentsExample extends JFrame {
  public SwingComponentsExample() {
    setTitle("Swing Components Example");
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLayout(new FlowLayout());
    JLabel label = new JLabel("Enter your name:");
    add(label);
    JTextField textField = new JTextField(15);
    add(textField);
    JButton button = new JButton("Submit");
    add(button);
    JCheckBox checkBox = new JCheckBox("I agree to the terms and conditions");
    add(checkBox);
    JTextArea textArea = new JTextArea(5, 20);
    textArea.setLineWrap(true);
    JScrollPane scrollPane = new JScrollPane(textArea);
    add(scrollPane);
    setSize(300, 300);
    setVisible(true);
  }
  public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        new SwingComponentsExample();
      }
    });
  }
}
```



10. Write a program to show implementation event handling using FocusListener.

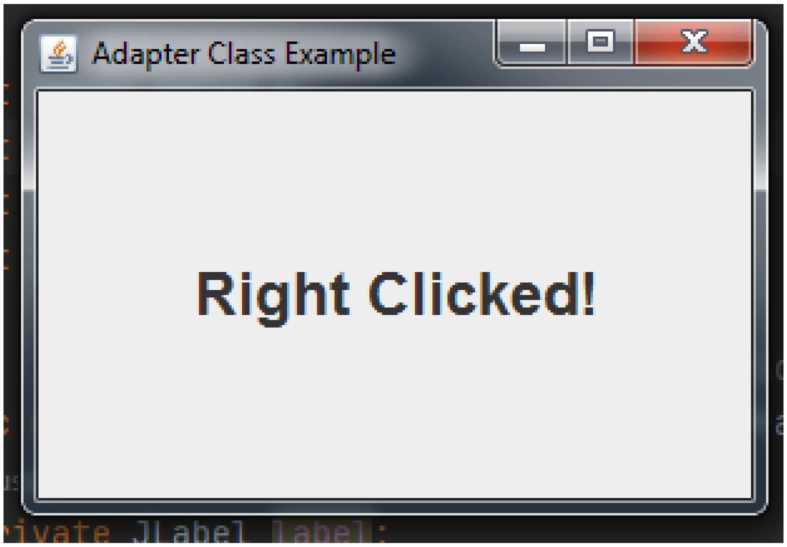
```
package lab1;
import javax.swing.*;
import java.awt.*;
import java.awt.event.FocusEvent;
import java.awt.event.FocusListener;
public class FocusListenerExample extends JFrame {
    public FocusListenerExample() {
        setTitle("Focus Listener Example");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new FlowLayout());
        JTextField textField1 = new JTextField(15);
        JTextField textField2 = new JTextField(15);
        FocusListener focusListener = new FocusListener() {
            @Override
            public void focusGained(FocusEvent e) {
                JTextField textField = (JTextField) e.getSource();
                textField.setBackground(Color.YELLOW);
            }
            @Override
            public void focusLost(FocusEvent e) {
                JTextField textField = (JTextField) e.getSource();
                textField.setBackground(Color.WHITE);
            }
        };
        textField1.addFocusListener(focusListener);
        textField2.addFocusListener(focusListener);
        add(new JLabel("Text Field 1:"));
        add(textField1);
        add(new JLabel("Text Field 2:"));
        add(textField2);
        setSize(300, 200);
        setVisible(true);
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new FocusListenerExample();
            }
        });
    }
}
```



11. Write a program to show implementation of Adapter Class.

```
package lab1.ques10;
import javax.swing.*;
import java.awt.*;
import java.awt.event.MouseAdapter;
import java.awt.event.MouseEvent;
//11.          Write a program to show implementation of Adapter Class.
public class AdapterClassExample extends JFrame {
   private JLabel label;
   public AdapterClassExample() {
      setTitle("Adapter Class Example");
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      label = new JLabel("Click me!");
      label.setHorizontalAlignment(SwingConstants.CENTER);
      label.setFont(new Font("Arial", Font.BOLD, 24));
      add(label);
      MouseAdapter mouseAdapter = new MouseAdapter() {
         @Override
         public void mouseClicked(MouseEvent e) {
            if (e.getButton() == MouseEvent.BUTTON1) {
               label.setText("Left Clicked!");
            } else if (e.getButton() == MouseEvent.BUTTON3) {
               label.setText("Right Clicked!");
            }
         }
      };
      label.addMouseListener(mouseAdapter);
      setSize(300, 200);
      setVisible(true);
   }
```

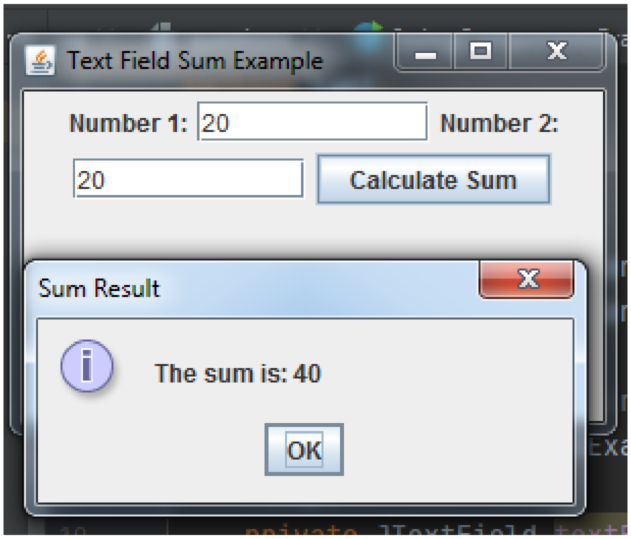
```
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new AdapterClassExample();
            }
        });
    }
}
```



```
Right Clicked!
```

12. Write a program to read two numbers from text field and sum and display in dialog box when button pressed.

```
package lab1;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
public class TextFieldSumExample extends JFrame {
    private JTextField textField1;
    private JTextField textField2;
    public TextFieldSumExample() {
        setTitle("Text Field Sum Example");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new FlowLayout());
        textField1 = new JTextField(10);
        textField2 = new JTextField(10);
        JButton sumButton = new JButton("Calculate Sum");
        ActionListener sumButtonListener = new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                String text1 = textField1.getText();
                String text2 = textField2.getText();
        try {
          int num1 = Integer.parseInt(text1);
          int num2 = Integer.parseInt(text2);
          int sum = num1 + num2;
          JOptionPane.showMessageDialog(TextFieldSumExample.this,
              "The sum is: " + sum, "Sum Result",
JOptionPane.INFORMATION_MESSAGE);
        } catch (NumberFormatException ex) {
          JOptionPane.showMessageDialog(TextFieldSumExample.this,
              "Invalid input! Please enter valid numbers.", "Error",
JOptionPane.ERROR_MESSAGE);
        }
      }
    };
    sumButton.addActionListener(sumButtonListener);
    add(new JLabel("Number 1:"));
    add(textField1);
    add(new JLabel("Number 2:"));
    add(textField2);
    add(sumButton);
    setSize(300, 200);
    setVisible(true);
  }
  public static void main(String[] args) {
    SwingUtilities.invokeLater(new Runnable() {
      public void run() {
        new TextFieldSumExample();
      }
    });
  }
}
```



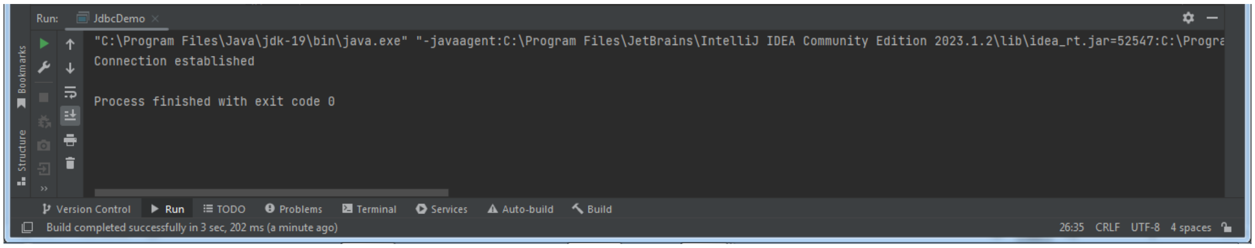
Number 1: 20
Number 2: 20

The sum is: 40

## LAB 2

1. Write a program to show database connectivity using JDBC.

```java
package lab2.ques1;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.sql.Statement;
//1.Write a program to show database connectivity using JDBC.
public class JdbcDemo {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/java";
        String username = "root";
        String password = "";
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection connection = DriverManager.getConnection(url, username, password);
            Statement statement = connection.createStatement();
            String query = "SELECT * FROM employees";
            if (!statement.isClosed()) {
                System.out.println("Connection established");
            } else {
                System.out.println("Connection not established");
            }
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

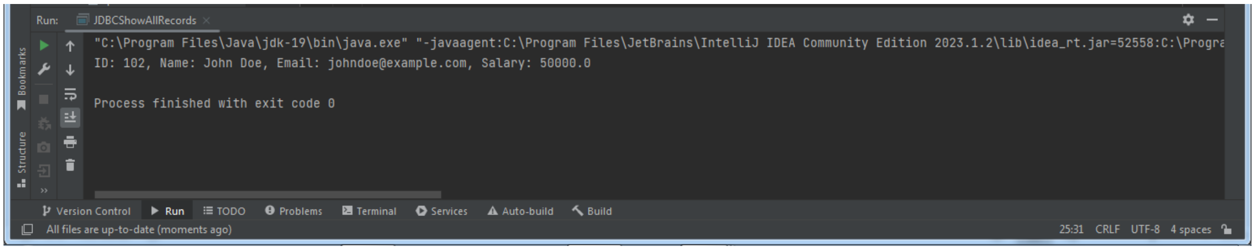


```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.1.2\lib\idea_rt.jar=52547:C:\Progra
Connection established

Process finished with exit code 0
```

Build completed successfully in 3 sec, 202 ms (a minute ago)

2. Write a program to show all the records from employee table.

```
package lab2.ques2;
import java.sql.*;
public class JDBCShowAllRecords {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/java";
        String username = "root";
        String password = "";
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection connection = DriverManager.getConnection(url, username, password);
            Statement statement = connection.createStatement();
            String query = "SELECT * FROM employees";
            ResultSet resultSet = statement.executeQuery(query);
            while (resultSet.next()) {
                int id = resultSet.getInt("id");
                String name = resultSet.getString("name");
                String email = resultSet.getString("email");
                double salary = resultSet.getDouble("salary");
                System.out.println("ID: " + id + ", Name: " + name + ", Email: " + email + ", Salary: 
" + salary);
            }
            resultSet.close();
            statement.close();
            connection.close();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```

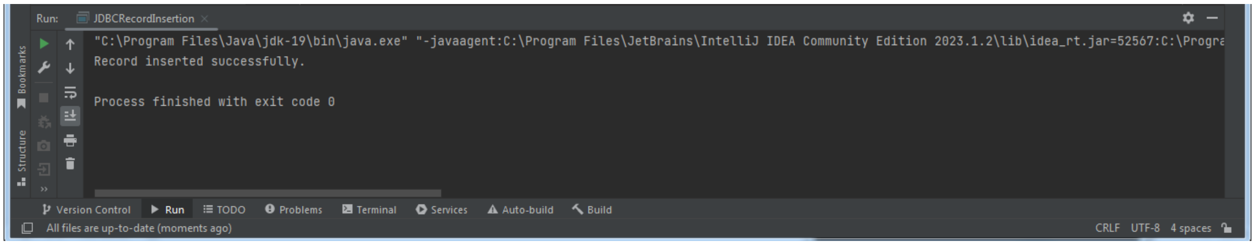


```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.1.2\lib\idea_rt.jar=52558:C:\Progra
ID: 102, Name: John Doe, Email: johndoe@example.com, Salary: 50000.0

Process finished with exit code 0
```

3. Write a program to show record insertion in employee table using PreparedStatement.

```
package lab2.ques3;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.SQLException;
```

```
public class JDBCRecordInsertion {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/java";
        String username = "root";
        String password = "";
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection connection = DriverManager.getConnection(url, username, password);
            String query = "INSERT INTO employees (name, email, salary) VALUES ( ?, ?, ?)";
            PreparedStatement preparedStatement = connection.prepareStatement(query);
            preparedStatement.setString(1, "John Doe");
            preparedStatement.setString(2, "johndoe@example.com");
            preparedStatement.setDouble(3, 50000.0);
            int rowsAffected = preparedStatement.executeUpdate();
            if (rowsAffected > 0) {
                System.out.println("Record inserted successfully.");
            } else {
                System.out.println("Failed to insert record.");
            }
            preparedStatement.close();
            connection.close();
        } catch (ClassNotFoundException e) {
            e.printStackTrace();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```



```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.1.2\lib\idea_rt.jar=52567:C:\Progra
Record inserted successfully.

Process finished with exit code 0
```

All files are up-to-date (moments ago)

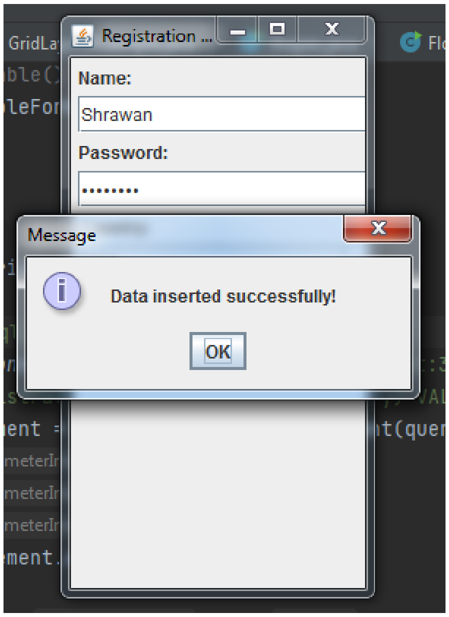
CRLF UTF-8 4 spaces

4. Write a program to create registration form using different JComponents. When the user clicks submit button, the entered data should store in database and on click of show button the data should be shown in JTable.

```
package lab2.ques4;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;
public class JdbcJTableForm extends JFrame {
   private JTextField nameField;
   private JPasswordField passwordField;
   private JComboBox<String> countryComboBox;
   private JButton submitButton;
   private JButton showButton;
   private JTable dataTable;
   private Connection connection;
   private Statement statement;
   public JdbcJTableForm() {
      JPanel panel = new JPanel();
      panel.setLayout(new FlowLayout(FlowLayout.LEFT));
      setTitle("Registration Form");
      setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      setSize(400, 400);
      JLabel nameLabel = new JLabel("Name:");
      nameField = new JTextField(20);
      nameField.setPreferredSize(new Dimension(200, 25));
      JLabel passwordLabel = new JLabel("Password:");
      passwordField = new JPasswordField(20);
      passwordField.setPreferredSize(new Dimension(200, 25));
      JLabel countryLabel = new JLabel("Country:");
      String[] countries = {"USA", "Canada", "UK"};
      countryComboBox = new JComboBox<>(countries);
      countryComboBox.setPreferredSize(new Dimension(200, 25));
      submitButton = new JButton("Submit");
      showButton = new JButton("Show Data");
      submitButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            String name = nameField.getText();
            String password = new String(passwordField.getPassword());
            String country = (String) countryComboBox.getSelectedItem();
            insertData(name, password, country);
         }
      });

      showButton.addActionListener(new ActionListener() {
         @Override
         public void actionPerformed(ActionEvent e) {
            retrieveAndDisplayData();
         }
      });
      panel.add(nameLabel);
      panel.add(nameField);
      panel.add(passwordLabel);
      panel.add(passwordField);
```

```
        panel.add(countryLabel);
        panel.add(countryComboBox);
        panel.add(submitButton);
        panel.add(showButton);
        add(panel);
        setVisible(true);
    }
    public static void main(String[] args) {
        SwingUtilities.invokeLater(new Runnable() {
            public void run() {
                new JdbcJTableForm();
            }
        });
    }
    private void insertData(String name, String password, String country) {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/java",
"root", "");
            String query = "INSERT INTO registration (name, password, country) VALUES (?, ?,
?)";
            PreparedStatement preparedStatement = connection.prepareStatement(query);
            preparedStatement.setString(1, name);
            preparedStatement.setString(2, password);
            preparedStatement.setString(3, country);
            int rowsAffected = preparedStatement.executeUpdate();
            if (rowsAffected > 0) {
                JOptionPane.showMessageDialog(this, "Data inserted successfully!");
            } else {
                JOptionPane.showMessageDialog(this, "Failed to insert data.");
            }
            preparedStatement.close();
            connection.close();
        } catch (SQLException ex) {
            ex.printStackTrace();
        } catch (ClassNotFoundException e) {
            throw new RuntimeException(e);
        }
    }
    private void retrieveAndDisplayData() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            connection = DriverManager.getConnection("jdbc:mysql://localhost:3306/java",
"root", "");
            String query = "SELECT * FROM registration";
            statement = connection.createStatement();
            ResultSet resultSet = statement.executeQuery(query);
            DefaultTableModel tableModel = new DefaultTableModel();
```

```
                tableModel.addColumn("ID");
                tableModel.addColumn("Name");
                tableModel.addColumn("Password");
                tableModel.addColumn("Country");
                while (resultSet.next()) {
                    Object[] row = new Object[4];
                    row[0] = resultSet.getInt("id");
                    row[1] = resultSet.getString("name");
                    row[2] = resultSet.getString("password");
                    row[3] = resultSet.getString("country");
                    tableModel.addRow(row);
                }
                dataTable = new JTable(tableModel);
                JScrollPane scrollPane = new JScrollPane(dataTable);
                JFrame displayFrame = new JFrame("Data Display");
                displayFrame.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
                displayFrame.add(scrollPane);
                displayFrame.pack();
                displayFrame.setVisible(true);
                resultSet.close();
                statement.close();
                connection.close();
            } catch (SQLException ex) {
                ex.printStackTrace();
            } catch (ClassNotFoundException e) {
                throw new RuntimeException(e);
            }
        }
    }
```



Registration ...

Name:

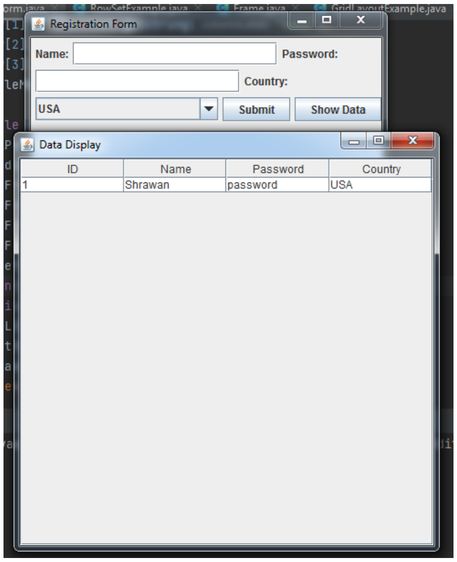
Shrawan

Password:

••••••••

Message

Data inserted successfully!

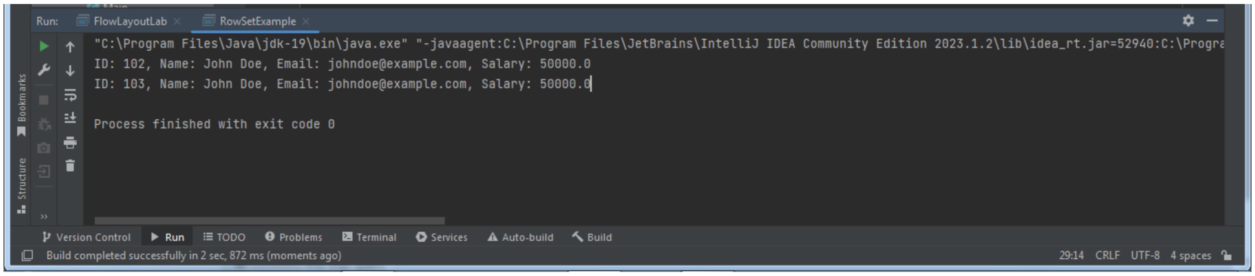


| ID | Name | Password | Country |
|---|---|---|---|
| 1 | Shrawan | password | USA |

5. Write a program to implement RowSet.

```
package lab2.ques5;
import javax.sql.rowset.JdbcRowSet;
import javax.sql.rowset.RowSetProvider;
import java.sql.SQLException;
public class RowSetExample {
   public static void main(String[] args) {
      String url = "jdbc:mysql://localhost:3306/java";
      String username = "root";
      String password = "";
      try {
         Class.forName("com.mysql.jdbc.Driver");
         JdbcRowSet rowSet = RowSetProvider.newFactory().createJdbcRowSet();
         rowSet.setUrl(url);
         rowSet.setUsername(username);
         rowSet.setPassword(password);
         rowSet.setCommand("SELECT * FROM employees");
         rowSet.execute();
         while (rowSet.next()) {
            int id = rowSet.getInt("id");
            String name = rowSet.getString("name");
            String email = rowSet.getString("email");
            double salary = rowSet.getDouble("salary");
            System.out.println("ID: " + id + ", Name: " + name + ", Email: " + email + ", Salary: 
" + salary);
         }
         rowSet.close();
      } catch (SQLException e) {
         e.printStackTrace();
      } catch (ClassNotFoundException e) {
         throw new RuntimeException(e);
```

```
        }
    }
}
```



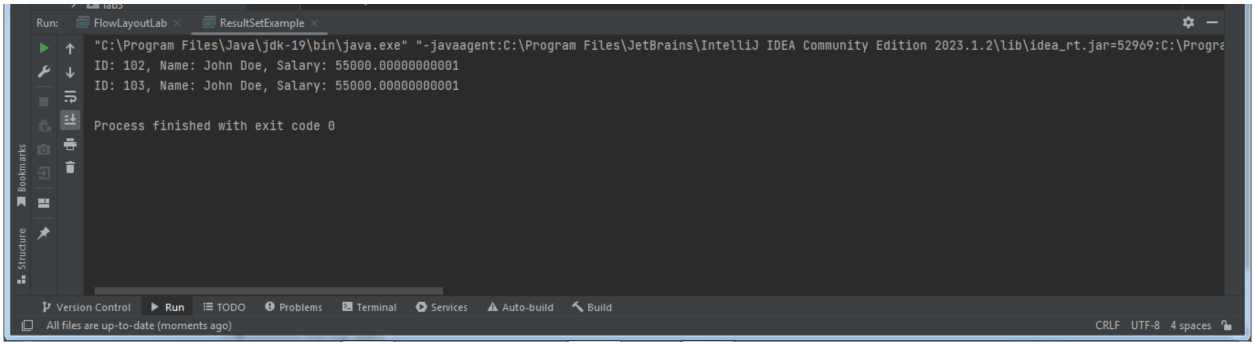
```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.1.2\lib\idea_rt.jar=52940:C:\Progra
ID: 102, Name: John Doe, Email: johndoe@example.com, Salary: 50000.0
ID: 103, Name: John Doe, Email: johndoe@example.com, Salary: 50000.0

Process finished with exit code 0
```

Build completed successfully in 2 sec, 872 ms (moments ago)

29:14 CRLF UTF-8 4 spaces

6. Write a program to implement updatable and scrollable ResultSet.

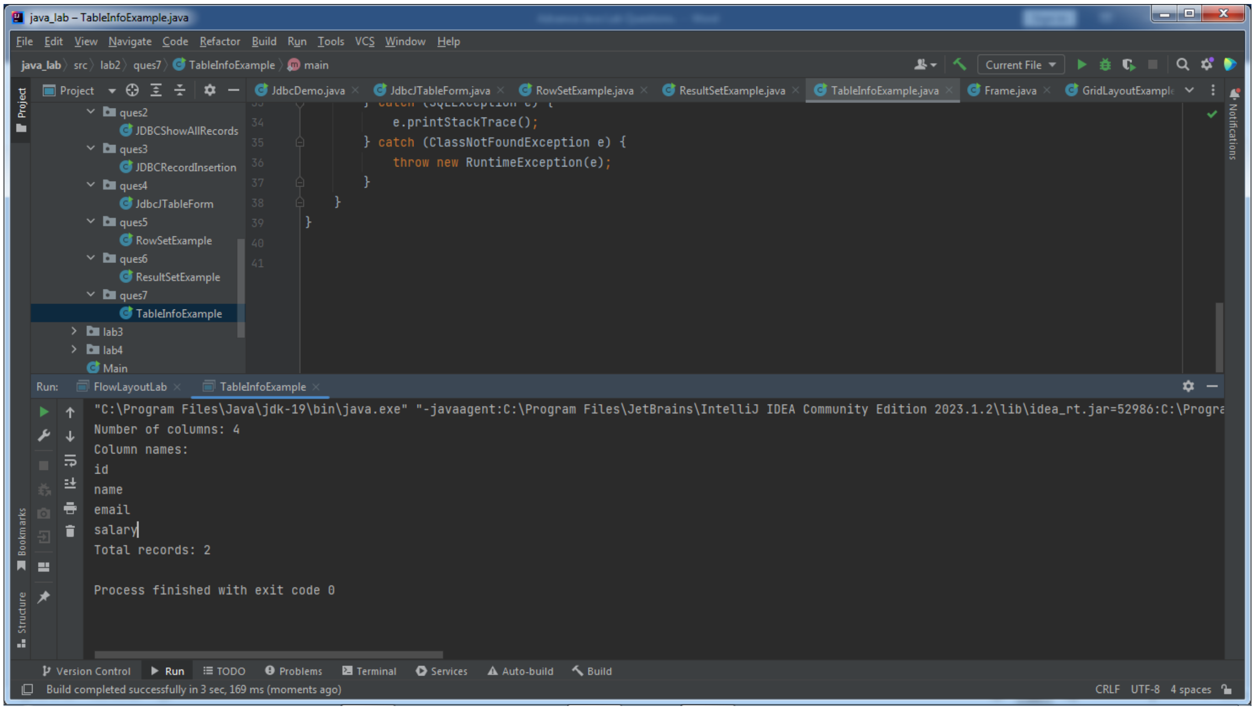
```
package lab2.ques6;
import java.sql.*;
public class ResultSetExample {
    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/java";
        String username = "root";
        String password = "";
        try {
            Class.forName("com.mysql.jdbc.Driver");
            Connection connection = DriverManager.getConnection(url, username, password);
            Statement statement =
connection.createStatement(ResultSet.TYPE_SCROLL_SENSITIVE,
ResultSet.CONCUR_UPDATABLE);
            String query = "SELECT * FROM employees";
            ResultSet resultSet = statement.executeQuery(query);
            while (resultSet.next()) {
                int id = resultSet.getInt("id");
                String name = resultSet.getString("name");
                double salary = resultSet.getDouble("salary");
                System.out.println("ID: " + id + ", Name: " + name + ", Salary: " + salary);
                resultSet.updateDouble("salary", salary * 1.1); // Increase the salary by 10%
                resultSet.updateRow(); // Save the updated row to the database
            }
            resultSet.close();
            statement.close();
            connection.close();
        } catch (SQLException e) {
            e.printStackTrace();
        } catch (ClassNotFoundException e) {
            throw new RuntimeException(e);
        }
    }
}
```



```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.1.2\lib\idea_rt.jar=52969:C:\Progra
ID: 102, Name: John Doe, Salary: 55000.00000000001
ID: 103, Name: John Doe, Salary: 55000.00000000001

Process finished with exit code 0
```

Run: FlowLayoutLab ResultSetExample
Version Control Run TODO Problems Terminal Services Auto-build Build
All files are up-to-date (moments ago)
CRLF UTF-8 4 spaces

7. Write a program to show number of columns, name of columns and total record from a specified table.



```
                e.printStackTrace();
            } catch (ClassNotFoundException e) {
                throw new RuntimeException(e);
            }
        }
    }
```

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2023.1.2\lib\idea_rt.jar=52986:C:\Progra
Number of columns: 4
Column names:
id
name
email
salary
Total records: 2

Process finished with exit code 0
```

java_lab – TableInfoExample.java
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help
java_lab src lab2 ques7 TableInfoExample main
Current File
Project
JdbcDemo.java JdbcJTableForm.java RowSetExample.java ResultSetExample.java TableInfoExample.java Frame.java GridLayoutExample
ques2 JDBCShowAllRecords ques3 JDBCRecordInsertion ques4 JdbcJTableForm ques5 RowSetExample ques6 ResultSetExample ques7 TableInfoExample lab3 lab4 Main
Run: FlowLayoutLab TableInfoExample
Version Control Run TODO Problems Terminal Services Auto-build Build
Build completed successfully in 3 sec, 169 ms (moments ago)
CRLF UTF-8 4 spaces

## LAB 3

1. Create a student javabeans with all possible properties and use it in another class.

```
package lab3.ques1;
//1.Create a student javabeans with all possible properties and use it in another class.
class StudentTest {
    public static void main(String[] args) {
        Student student = new Student();
        student.setId(1);
        student.setName("John Doe");
        student.setAge(18);
        student.setGrade("A");
        System.out.println("Student ID: " + student.getId());
        System.out.println("Student Name: " + student.getName());
        System.out.println("Student Age: " + student.getAge());
        System.out.println("Student Grade: " + student.getGrade());
    }
}
public class Student {
    private int id;
    private String name;
    private int age;
    private String grade;

    public Student() {
    }
    public Student(int id, String name, int age, String grade) {
        this.id = id;
        this.name = name;
        this.age = age;
        this.grade = grade;
    }
    public int getId() {
        return id;
    }
    public void setId(int id) {
        this.id = id;
    }
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    public String getGrade() {
        return grade;
    }
    public void setGrade(String grade) {
        this.grade = grade;
    }
}
```

2. Write a program to use propertyDescriptor in javabeans.

```
package lab3.ques2;
import lab3.ques1.Student;
import java.beans.BeanInfo;
import java.beans.Introspector;
import java.beans.PropertyDescriptor;
public class BeanPropertyDescriptorExample {
   public static void main(String[] args) {
      Student student = new Student();
      try {
         BeanInfo beanInfo = Introspector.getBeanInfo(Student.class);
         PropertyDescriptor[] propertyDescriptors = beanInfo.getPropertyDescriptors();
         for (PropertyDescriptor propertyDescriptor : propertyDescriptors) {
            if ("id".equals(propertyDescriptor.getName())) {
               propertyDescriptor.getWriteMethod().invoke(student, 1);
            } else if ("name".equals(propertyDescriptor.getName())) {
               propertyDescriptor.getWriteMethod().invoke(student, "John Doe");
            } else if ("age".equals(propertyDescriptor.getName())) {
               propertyDescriptor.getWriteMethod().invoke(student, 18);
            } else if ("grade".equals(propertyDescriptor.getName())) {
               propertyDescriptor.getWriteMethod().invoke(student, "A");
            }
         }
         for (PropertyDescriptor propertyDescriptor : propertyDescriptors) {
            Object value = propertyDescriptor.getReadMethod().invoke(student);
            System.out.println(propertyDescriptor.getName() + ": " + value);
         }
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
}
```

3. Write a program to use eventDescriptor in javabeans.

```
package lab3.ques2;
import lab3.ques1.Student;
import java.beans.BeanInfo;
import java.beans.Introspector;
import java.beans.PropertyDescriptor;
public class BeanPropertyDescriptorExample {
   public static void main(String[] args) {
      Student student = new Student();
      try {
         BeanInfo beanInfo = Introspector.getBeanInfo(Student.class);
         PropertyDescriptor[] propertyDescriptors = beanInfo.getPropertyDescriptors();
         for (PropertyDescriptor propertyDescriptor : propertyDescriptors) {
            if ("id".equals(propertyDescriptor.getName())) {
               propertyDescriptor.getWriteMethod().invoke(student, 1);
            } else if ("name".equals(propertyDescriptor.getName())) {
               propertyDescriptor.getWriteMethod().invoke(student, "John Doe");
            } else if ("age".equals(propertyDescriptor.getName())) {
               propertyDescriptor.getWriteMethod().invoke(student, 18);
            } else if ("grade".equals(propertyDescriptor.getName())) {
               propertyDescriptor.getWriteMethod().invoke(student, "A");
            }
         }
         for (PropertyDescriptor propertyDescriptor : propertyDescriptors) {
            Object value = propertyDescriptor.getReadMethod().invoke(student);
            System.out.println(propertyDescriptor.getName() + ": " + value);
         }
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
}
```

4. Write a program to use InfoBean in javabeans.

```
package lab3.ques4;
//4. Write a program to use InfoBean in javabeans.
public class PersonInfoBean {
   private String name;
   private int age;
   public String getInfo() {
      return "Name: " + name + "\nAge: " + age;
   }
   public void setName(String name) {
      this.name = name;
   }
   public void setAge(int age) {
      this.age = age;
   }
}
class InfoBeanExample {
   public static void main(String[] args) {
      PersonInfoBean person = new PersonInfoBean();
      person.setName("John Doe");
      person.setAge(25);
      String info = person.getInfo();
      System.out.println(info);
   }
}
```

5. Write a program to implement client-server application using RMI.

```
package lab3.ques5;
import java.rmi.Remote;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;
interface RemoteService extends Remote {
  String sayHello() throws RemoteException;
}
class RMIClient {
  public static void main(String[] args) {
    try {
      Registry registry = LocateRegistry.getRegistry("localhost", 1099);
      RemoteService remoteService = (RemoteService)
registry.lookup("RemoteService");

      String message = remoteService.sayHello();
      System.out.println("Server says: " + message);
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
class RemoteServiceImpl extends UnicastRemoteObject implements RemoteService {
  protected RemoteServiceImpl() throws RemoteException {
    super();
  }

  public String sayHello() throws RemoteException {
    return "Hello from server!";
  }
}
class RMIServer {
  public static void main(String[] args) {
    try {
      RemoteService remoteService = new RemoteServiceImpl();

      Registry registry = LocateRegistry.createRegistry(1099);
      registry.bind("RemoteService", remoteService);

      System.out.println("Server is running...");
    } catch (Exception e) {
      e.printStackTrace();
    }
  }
}
```

6. Write a program using RMI and swing. Only use add, subtract, multiply and divide.

```
package lab3.ques6;
import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.rmi.Remote;
import java.rmi.RemoteException;
import java.rmi.registry.LocateRegistry;
import java.rmi.registry.Registry;
import java.rmi.server.UnicastRemoteObject;
interface RemoteCalculator extends Remote {
    double add(double num1, double num2) throws RemoteException;
    double subtract(double num1, double num2) throws RemoteException;
    double multiply(double num1, double num2) throws RemoteException;
    double divide(double num1, double num2) throws RemoteException;
}
class RemoteCalculatorImpl extends UnicastRemoteObject implements RemoteCalculator
{
    protected RemoteCalculatorImpl() throws RemoteException {
        super();
    }
    public double add(double num1, double num2) throws RemoteException {
        return num1 + num2;
    }
    public double subtract(double num1, double num2) throws RemoteException {
        return num1 - num2;
    }
    public double multiply(double num1, double num2) throws RemoteException {
        return num1 * num2;
    }
    public double divide(double num1, double num2) throws RemoteException {
        return num1 / num2;
    }
}
class RMIServerSwing {
    public static void main(String[] args) {
        try {
            RemoteCalculator remoteCalculator = new RemoteCalculatorImpl();

            Registry registry = LocateRegistry.createRegistry(1099);
            registry.bind("RemoteCalculator", remoteCalculator);

            System.out.println("Server is running...");
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
class RMIClientSwing extends JFrame implements ActionListener {
  private final JButton addButton;
  private final JButton subtractButton;
  private final JButton multiplyButton;
  private final JButton divideButton;
  private final JTextField num1Field;
  private final JTextField num2Field;
  private final JTextField resultField;
  private RemoteCalculator remoteCalculator;
  public RMIClientSwing() {
    setTitle("RMI Calculator");
    setSize(300, 200);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    setLocationRelativeTo(null);
    JPanel panel = new JPanel();
    panel.setLayout(new GridLayout(4, 2));
    panel.add(new JLabel("Number 1:"));
    num1Field = new JTextField();
    panel.add(num1Field);
    panel.add(new JLabel("Number 2:"));
    num2Field = new JTextField();
    panel.add(num2Field);
    addButton = new JButton("Add");
    addButton.addActionListener(this);
    panel.add(addButton);
    subtractButton = new JButton("Subtract");
    subtractButton.addActionListener(this);
    panel.add(subtractButton);
    multiplyButton = new JButton("Multiply");
    multiplyButton.addActionListener(this);
    panel.add(multiplyButton);
    divideButton = new JButton("Divide");
    divideButton.addActionListener(this);
    panel.add(divideButton);
    panel.add(new JLabel("Result:"));
    resultField = new JTextField();
    resultField.setEditable(false);
    panel.add(resultField);
    add(panel);
  }
  public static void main(String[] args) {
    try {
      Registry registry = LocateRegistry.getRegistry("localhost", 1099);
      RemoteCalculator remoteCalculator = (RemoteCalculator)
registry.lookup("RemoteCalculator");

      RMIClientSwing client = new RMIClientSwing();
      client.setRemoteCalculator(remoteCalculator);
```

```
         client.setVisible(true);
      } catch (Exception e) {
         e.printStackTrace();
      }
   }
   public void actionPerformed(ActionEvent e) {
      try {
         double num1 = Double.parseDouble(num1Field.getText());
         double num2 = Double.parseDouble(num2Field.getText());

         if (e.getSource() == addButton) {
            double result = remoteCalculator.add(num1, num2);
            resultField.setText(String.valueOf(result));
         } else if (e.getSource() == subtractButton) {
            double result = remoteCalculator.subtract(num1, num2);
            resultField.setText(String.valueOf(result));
         } else if (e.getSource() == multiplyButton) {
            double result = remoteCalculator.multiply(num1, num2);
            resultField.setText(String.valueOf(result));
         } else if (e.getSource() == divideButton) {
            double result = remoteCalculator.divide(num1, num2);
            resultField.setText(String.valueOf(result));
         }
      } catch (NumberFormatException ex) {
         JOptionPane.showMessageDialog(this, "Invalid input! Please enter numeric
values.");
      } catch (RemoteException ex) {
         JOptionPane.showMessageDialog(this, "Error communicating with the server.");
      }
   }
   public void setRemoteCalculator(RemoteCalculator remoteCalculator) {
      this.remoteCalculator = remoteCalculator;
   }
}
```

## LAB 4

1. Create a servlet/jsp that display one text box and one button in web browser and displays entered number's factorial when button is clicked.

```
<html>
<head>
   <title>Factorial Calculator</title>
</head>
<body>
   <h2>Factorial Calculator</h2>
   <form action="factorialServlet" method="post">
      Enter a number:
        <input type="text" name="number" required>
        <input type="submit" value="Calculate Factorial">
      </form>
    </body>
    </html>
```

```
        package lab4.ques1;
        import javax.servlet.ServletException;
        import javax.servlet.http.HttpServlet;
        import javax.servlet.http.HttpServletRequest;
        import javax.servlet.http.HttpServletResponse;
        import java.io.IOException;
        public class FactorialServlet extends HttpServlet {
          protected void doPost(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
            int number = Integer.parseInt(request.getParameter("number"));
            int factorial = calculateFactorial(number);
            request.setAttribute("factorial", factorial);
            request.getRequestDispatcher("result.jsp").forward(request, response);
          }
          private int calculateFactorial(int number) {
            if (number == 0 || number == 1) {
              return 1;
            } else {
              return number * calculateFactorial(number - 1);
            }
          }
}
```

2. Create a HTML form to save company name, city and zip code in database.

```
    <%@ page import="java.sql.*" %>
    <%
      String companyName = request.getParameter("companyName");
      String city = request.getParameter("city");
      String zipCode = request.getParameter("zipCode");
      String url = "jdbc:mysql://localhost:3306/your_database_name";
      String username = "your_username";
      String password = "your_password";
      Connection conn = null;
      PreparedStatement pstmt = null;
      try {
        Class.forName("com.mysql.jdbc.Driver");
        conn = DriverManager.getConnection(url, username, password);
        String sql = "INSERT INTO company (company_name, city, zip_code) VALUES (?, ?, ?)";
        pstmt = conn.prepareStatement(sql);
        pstmt.setString(1, companyName);
        pstmt.setString(2, city);
        pstmt.setString(3, zipCode);
        pstmt.executeUpdate();
        pstmt.close();
        conn.close();
        response.sendRedirect("success.html"); // Redirect to a success page
      } catch (Exception e) {
        e.printStackTrace();
        response.sendRedirect("error.html"); // Redirect to an error page
      }
    %>
```

3. Write a program to show session tracking using all the possible ways.
   - Hidden form field
   - URL Rewriting
   - Cookies
   - Session

```
<!DOCTYPE html>
<html>
<head>
  <title>Session Tracking Example</title>
</head>
<body>
<h2>Session Tracking Example</h2>
<form action="hiddenFieldServlet" method="post">
  <label for="name">Name:</label>
  <input id="name" name="name" required type="text"><br><br>
  <input type="submit" value="Submit">
</form>
<br>
<a href="urlRewritingServlet?name=John">URL Rewriting</a>
<br><br>
<form action="cookieServlet" method="post">
  <label for="name">Name:</label>
  <input id="name" name="name" required type="text"><br><br>
  <input type="submit" value="Submit">
</form>
<br>
<form action="sessionServlet" method="post">
  <label for="name">Name:</label>
  <input id="name" name="name" required type="text"><br><br>
  <input type="submit" value="Submit">
</form>
</body>
</html>
```

```
package lab4.ques3;
```

```
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
public class CookieServlet extends HttpServlet {
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    String name = request.getParameter("name");
    Cookie cookie = new Cookie("name", name);
    response.addCookie(cookie);
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html><head><title>Cookie - Session Tracking</title></head><body>");
    out.println("<h2>Cookie - Session Tracking</h2>");
    out.println("<form action='displayDataServlet' method='post'>");
    out.println("<input type='submit' value='Display Data'>");
    out.println("</form>");
    out.println("</body></html>");
  }
}
```

```
package lab4.ques3;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.io.PrintWriter;
public class DisplayDataServlet extends HttpServlet {
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    HttpSession session = request.getSession();
    String name = (String) session.getAttribute("name");
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html><head><title>Display Data - Session Tracking</title></head><body>");
    out.println("<h2>Display Data - Session Tracking</h2>");
    out.println("<p>Name: " + name + "</p>");
    out.println("</body></html>");
  }
}
```

```
package lab4.ques3;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.io.PrintWriter;
public class HiddenFieldServlet extends HttpServlet {
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    String name = request.getParameter("name");
    HttpSession session = request.getSession();
    session.setAttribute("name", name);
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html><head><title>Hidden Field - Session Tracking</title></head><body>");
    out.println("<h2>Hidden Field - Session Tracking</h2>");
    out.println("<form action='displayDataServlet' method='post'>");
    out.println("<input type='hidden' name='name' value='" + name + "'>");
    out.println("<input type='submit' value='Display Data'>");
    out.println("</form>");
    out.println("</body></html>");
  }
}
```

```
package lab4.ques3;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import javax.servlet.http.HttpSession;
import java.io.IOException;
import java.io.PrintWriter;
public class URLRewritingServlet extends HttpServlet {
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    String name = request.getParameter("name");
    HttpSession session = request.getSession();
    session.setAttribute("name", name);
    response.setContentType("text/html");
    PrintWriter out = response.getWriter();
    out.println("<html><head><title>URL Rewriting - Session Tracking</title></head><body>");
    out.println("<h2>URL Rewriting - Session Tracking</h2>");
    out.println("<a href='displayDataServlet'>Display Data</a>");
    out.println("</body></html>");
  }
}
```

4. Write a servlet program to perform operation using include(), forward() and sendRedirect() method.

```
package lab4.ques4;
import javax.servlet.RequestDispatcher;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
public class OperationServlet extends HttpServlet {
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
        String operation = request.getParameter("operation");
        if (operation.equals("include")) {
            includeOperation(request, response);
        } else if (operation.equals("forward")) {
            forwardOperation(request, response);
        } else if (operation.equals("redirect")) {
            redirectOperation(request, response);
        } else {
            response.setContentType("text/html");
            PrintWriter out = response.getWriter();
            out.println("<html><head><title>Operation</title></head><body>");
            out.println("<h2>Invalid operation specified!</h2>");
            out.println("</body></html>");
        }
    }
    private void includeOperation(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        out.println("<html><head><title>Include Operation</title></head><body>");
        out.println("<h2>Included Content</h2>");
        RequestDispatcher rd = request.getRequestDispatcher("/includedServlet");
        rd.include(request, response);
        out.println("<h2>Include Operation Completed</h2>");
        out.println("</body></html>");
    }
    private void forwardOperation(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        RequestDispatcher rd = request.getRequestDispatcher("/forwardedServlet");
        rd.forward(request, response);
    }
    private void redirectOperation(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.sendRedirect(request.getContextPath() + "/redirectedServlet");
    }
}
```

5. Write a servlet program to read initialization parameters from web.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<web-app xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
         xmlns="http://java.sun.com/xml/ns/javaee"
         xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
                             http://java.sun.com/xml/ns/javaee/web-app_4_0.xsd"
         version="4.0">
    <servlet>
        <servlet-name>InitializationServlet</servlet-name>
        <servlet-class>com.example.InitializationServlet</servlet-class>
        <init-param>
            <param-name>databaseUrl</param-name>
            <param-value>jdbc:mysql://localhost:3306/mydb</param-value>
        </init-param>
        <init-param>
            <param-name>username</param-name>
            <param-value>admin</param-value>
        </init-param>
        <init-param>
            <param-name>password</param-name>
            <param-value>password123</param-value>
        </init-param>
    </servlet>
    <servlet-mapping>
        <servlet-name>InitializationServlet</servlet-name>
        <url-pattern>/init</url-pattern>
    </servlet-mapping>
</web-app>
```

```
package lab4.ques5;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.io.PrintWriter;
public class InitializationServlet extends HttpServlet {
    private String databaseUrl;
```

```
        private String username;
        private String password;
        @Override
        public void init() throws ServletException {
          // Read the initialization parameters from web.xml
          databaseUrl = getServletContext().getInitParameter("databaseUrl");
          username = getServletContext().getInitParameter("username");
          password = getServletContext().getInitParameter("password");
        }
        @Override
        protected void doGet(HttpServletRequest request, HttpServletResponse response)
            throws ServletException, IOException {
          response.setContentType("text/html");
          PrintWriter out = response.getWriter();
          out.println("<html><head><title>Initialization
      Parameters</title></head><body>");
          out.println("<h2>Initialization Parameters</h2>");
          out.println("<p>Database URL: " + databaseUrl + "</p>");
          out.println("<p>Username: " + username + "</p>");
          out.println("<p>Password: " + password + "</p>");
          out.println("</body></html>");
        }
}
```

6. Write a JSP program to write *"KCT"*10 times.

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
  <title>Print KCT 10 Times</title>
</head>
<body>
  <h2>Printing "KCT" 10 Times</h2>
  <%
    for (int i = 1; i <= 10; i++) {
      out.println("KCT ");
    }
  %>
</body>
</html>
```

7. Write a JSP program to perform HTML form data processing.

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
```

```
  <title>HTML Form Data Processing</title>
</head>
<body>
  <h2>HTML Form Data Processing</h2>
  <%-- Check if the form is submitted --%>
  <% if (request.getMethod().equalsIgnoreCase("POST")) {
    // Retrieve form data
    String name = request.getParameter("name");
    String email = request.getParameter("email");
    String message = request.getParameter("message");
    // Perform data processing or validation
    // For this example, we simply display the submitted data
  %>
  <p>Submitted Data:</p>
  <p>Name: <%= name %></p>
  <p>Email: <%= email %></p>
  <p>Message: <%= message %></p>
  <% } %>
  <form method="post" action="">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required><br><br>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required><br><br>

    <label for="message">Message:</label><br>
    <textarea id="message" name="message" rows="4" cols="30"
required></textarea><br><br>

    <input type="submit" value="Submit">
  </form>
</body>
</html>
```

8. Write a JSP program to perform Exception Handling.

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
  <title>Exception Handling</title>
</head>
<body>
  <h2>Exception Handling</h2>

  <%-- Perform some operation that may throw an exception --%>
  <%
    try {
      // Simulating an exception by dividing by zero
      int result = 10 / 0;
      out.println("Result: " + result);
    } catch (ArithmeticException e) {
      out.println("<p>An error occurred: " + e.getMessage() + "</p>");
    }
  %>
</body>
</html>
```

9. Write a JSP program to perform to read value from textbox and identifies whether it’s odd or even.

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
  <title>Odd or Even Number</title>
</head>
<body>
  <h2>Odd or Even Number</h2>

  <%-- Check if the form is submitted --%>
  <% if (request.getMethod().equalsIgnoreCase("POST")) {
    // Retrieve form data
    int number = Integer.parseInt(request.getParameter("number"));
    String result = number % 2 == 0 ? "even" : "odd";
  %>
  <p>The number <%= number %> is <%= result %>.</p>
  <% } %>

  <form method="post" action="">
    <label for="number">Enter a number:</label>
    <input type="text" id="number" name="number" required><br><br>

    <input type="submit" value="Check">
  </form>
</body>
</html>
```

10. Create a contact manager application using servlet and JSP.

```
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
  <title>Contact Manager</title>
</head>
```

```
<body>
  <h2>Contact Manager</h2>

  <%-- Display contacts --%>
  <% for (Contact contact : contacts) { %>
    <p>Name: <%= contact.getName() %></p>
    <p>Email: <%= contact.getEmail() %></p>
    <p>Phone: <%= contact.getPhone() %></p>
    <hr>
  <% } %>
  <%-- Add contact form --%>
  <h3>Add Contact</h3>
  <form method="post" action="contact">
    <label for="name">Name:</label>
    <input type="text" id="name" name="name" required><br><br>

    <label for="email">Email:</label>
    <input type="email" id="email" name="email" required><br><br>

    <label for="phone">Phone:</label>
    <input type="text" id="phone" name="phone" required><br><br>

    <input type="submit" value="Add Contact">
  </form>
</body>
</html>
```

```
package lab4.ques10;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
import java.io.IOException;
import java.util.ArrayList;
import java.util.List;
@WebServlet("/contact")
public class ContactManagerServlet extends HttpServlet {
  private List<Contact> contacts = new ArrayList<>();
  @Override
  protected void doGet(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
    request.setAttribute("contacts", contacts);
    request.getRequestDispatcher("contact.jsp").forward(request, response);
  }
  @Override
  protected void doPost(HttpServletRequest request, HttpServletResponse response)
      throws ServletException, IOException {
        String name = request.getParameter("name");
        String email = request.getParameter("email");
        String phone = request.getParameter("phone");
        Contact contact = new Contact(name, email, phone);
        contacts.add(contact);
        response.sendRedirect(request.getContextPath() + "/contact");
      }
    }
```

```
    package lab4.ques10;
    public class Contact {
      private String name;
      private String email;
      private String phone;
      public Contact(String name, String email, String phone) {
        this.name = name;
        this.email = email;
        this.phone = phone;
      }
      public String getName() {
        return name;
      }
      public void setName(String name) {
        this.name = name;
      }
      public String getEmail() {
        return email;
      }
      public void setEmail(String email) {
        this.email = email;
      }
      public String getPhone() {
        return phone;
      }
      public void setPhone(String phone) {
        this.phone = phone;
      }
}.
```