1. Bouncing Ball

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit this template
 */
package bouncingballdemo;

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

/**
 *
 * @author preety tilwani
 */
public class BouncingBallDemo extends JPanel {
    private int x = 50, y = 50;
    private int dx = 3, dy = 3;
    private final int BALL_SIZE = 30;

    public BouncingBallDemo(){
        Timer timer = new Timer(10, new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                x += dx;
                y += dy;

                if (x < 0 || x + BALL_SIZE > getWidth()) dx = -dx;
                if (y < 0 || y + BALL_SIZE > getHeight()) dy = -dy;

                repaint();
            }
        });
        timer.start();
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);
        g.setColor(Color.BLUE);
        g.fillOval(x, y, BALL_SIZE, BALL_SIZE);
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("Bouncing Ball");
        BouncingBallDemo panel = new BouncingBallDemo();
        frame.add(panel);
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

2. Moving Car Animation

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt
to change this license
 * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Main.java to edit
this template
 */
package carmovingdemo;

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

/**
 *
 * @author preety tilwani
 */
public class CarMovingDemo extends JPanel {
    private int x = -100;

    public CarMovingDemo() {
        Timer timer = new Timer(50, new ActionListener() {
            @Override
            public void actionPerformed(ActionEvent e) {
                x += 5;
                if (x > getWidth())
                    x = -100;
                repaint();
            }
        });
        timer.start();
    }

    @Override
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);

        g.setColor(Color.RED);
        g.fillRect(x, 100, 100, 50);
        g.fillRect(x + 20, 80, 60, 30);

        g.setColor(Color.BLACK);
        g.fillOval(x + 10, 140, 20, 20);
        g.fillOval(x + 70, 140, 20, 20);
    }

    public static void main(String[] args) {
        JFrame frame = new JFrame("Car Animation");
        CarMovingDemo panel = new CarMovingDemo();
        frame.add(panel);
        frame.setSize(400, 300);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }
}
```

3. Javabeans Demo student class

```
package studentlistdemo;


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


// Student class
class Student {
    private int rollNumber;
    private String name;
    private int marks1, marks2, marks3;


    public Student(int rollNumber, String name, int marks1, int marks2, int marks3) {
        this.rollNumber = rollNumber;
        this.name = name;
        this.marks1 = marks1;
        this.marks2 = marks2;
        this.marks3 = marks3;
    }


    public int getRollNumber() {
```

```
    return rollNumber;
}

public String getName() {
    return name;
}

public int getMarks1() {
    return marks1;
}

public int getMarks2() {
    return marks2;
}

public int getMarks3() {
    return marks3;
}

// Override toString to display the name in the JList
@Override
public String toString() {
    return name;
}

// Method to get full student details
public String getDetails() {
    return "Roll No: " + rollNumber + "\n" +
           "Name: " + name + "\n" +
           "Marks1: " + marks1 + "\n" +
           "Marks2: " + marks2 + "\n" +
           "Marks3: " + marks3 + "\n" +
            "Total: " + (marks1 + marks2 + marks3);
    }
}

public class StudentListDemo {
public static void main(String[] args) {
  JFrame frame = new JFrame("Student JList Example");
      frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
      frame.setSize(400, 400);
      frame.setLayout(new BorderLayout());

      DefaultListModel<Student> studentListModel = new DefaultListModel<>();
      JList<Student> studentJList = new JList<>(studentListModel);
      JScrollPane scrollPane = new JScrollPane(studentJList);
      studentJList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);

      JPanel inputPanel = new JPanel(new GridLayout(5, 2, 5, 5));

      JTextField rollField = new JTextField();
      JTextField nameField = new JTextField();
      JTextField marks1Field = new JTextField();
      JTextField marks2Field = new JTextField();
      JTextField marks3Field = new JTextField();
      JButton addButton = new JButton("Add Student");

      inputPanel.add(new JLabel("Roll Number:"));
      inputPanel.add(rollField);
      inputPanel.add(new JLabel("Name:"));
      inputPanel.add(nameField);
      inputPanel.add(new JLabel("Marks1:"));
      inputPanel.add(marks1Field);
      inputPanel.add(new JLabel("Marks2:"));
    inputPanel.add(marks2Field);
    inputPanel.add(new JLabel("Marks3:"));
    inputPanel.add(marks3Field);

    frame.add(inputPanel, BorderLayout.NORTH);
    frame.add(scrollPane, BorderLayout.CENTER);
    frame.add(addButton, BorderLayout.SOUTH);

    // Button Action to add students to the JList
    addButton.addActionListener(new ActionListener() {
      @Override
      public void actionPerformed(ActionEvent e) {
        try {
          int roll = Integer.parseInt(rollField.getText());
          String name = nameField.getText();
          int marks1 = Integer.parseInt(marks1Field.getText());
          int marks2 = Integer.parseInt(marks2Field.getText());
          int marks3 = Integer.parseInt(marks3Field.getText());

          Student student = new Student(roll, name, marks1, marks2,
marks3);
          studentListModel.addElement(student);

          rollField.setText("");
          nameField.setText("");
          marks1Field.setText("");
          marks2Field.setText("");
          marks3Field.setText("");

        } catch (NumberFormatException ex) {
          JOptionPane.showMessageDialog(frame, "Please enter valid
input!", "Error", JOptionPane.ERROR_MESSAGE);
        }
```

```
            }
        });

        studentJList.addListSelectionListener(e -> {
            if (!e.getValueIsAdjusting()) {
                Student selectedStudent = studentJList.getSelectedValue();
                if (selectedStudent != null) {
                    JOptionPane.showMessageDialog(frame,
selectedStudent.getDetails(), "Student Details",
JOptionPane.INFORMATION_MESSAGE);
                }
            }
        });

        frame.setVisible(true);
    }

}
```

4. Dynamic JList

```
package dynamicjlistexample;
import javax.swing.*;
import java.awt.event.*;
public class DynamicJListExample {

    public static void main(String[] args) {
        JFrame frame = new JFrame("Dynamic JList Example");
        DefaultListModel<String> model = new DefaultListModel<>();
        JList<String> list = new JList<>(model);
        JScrollPane scrollPane = new JScrollPane(list);

        JTextField textField = new JTextField(15);
        JButton addButton = new JButton("Add");
        JButton removeButton = new JButton("Remove");

        addButton.addActionListener(e -> {
            if (!textField.getText().isEmpty()) {
                model.addElement(textField.getText());
                textField.setText("");
            }
        });

        removeButton.addActionListener(e -> {
            int selectedIndex = list.getSelectedIndex();
            if (selectedIndex != -1) {
                model.remove(selectedIndex);
            }
        });

        JPanel panel = new JPanel();
        panel.add(textField);
        panel.add(addButton);
        panel.add(removeButton);

        frame.add(scrollPane, "Center");
        frame.add(panel, "South");

        frame.setSize(400, 250);
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        frame.setVisible(true);
    }

}
```

5. Select data from table

```
package employeejdbcdemo;
import java.sql.*;
import java.util.logging.Level;
import java.util.logging.Logger;
/**
 *
 * @author preety tilwani
 */
public class EmployeeJDBCDemo {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        try {
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection connection =
DriverManager.getConnection("jdbc:mysql://localhost:3306/userdb","roo
t","");
            System.out.println(connection);
            Statement s = connection.createStatement();
            ResultSet resultSet = s.executeQuery("Select * from employee");
            while(resultSet.next()){
                System.out.println(resultSet.getInt(1)+"");
                System.out.println(resultSet.getString(2)+"");
                System.out.println(resultSet.getString(3)+"");
            }
        } catch (ClassNotFoundException ex) {

Logger.getLogger(EmployeeJDBCDemo.class.getName()).log(Level.SEV
ERE, null, ex);
        } catch (SQLException ex) {

Logger.getLogger(EmployeeJDBCDemo.class.getName()).log(Level.SEV
ERE, null, ex);
        }

    }

}
```

6. Insert data in table:

```
package jdbcuserdemo;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

/**
 *
 * @author preety tilwani
 */
public class JDBCUserDemo extends JFrame{

    private JTextField nameField, emailField,idField;
    private JButton saveButton;
    private JTable userTable;
    private DefaultTableModel tableModel;

    private static final String URL = 
"jdbc:mysql://localhost:3306/userdb";
    private static final String USER = "root";
    private static final String PASSWORD = "";

    public JDBCUserDemo() {
        setTitle("User Management System");
        setSize(500, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        JPanel inputPanel = new JPanel(new GridLayout(5, 1, 10, 10));
        idField = new JTextField();
        nameField = new JTextField();
        emailField = new JTextField();
        saveButton = new JButton("Save");

        inputPanel.add(new JLabel("id:"));
        inputPanel.add(idField);
        inputPanel.add(new JLabel("Name:"));
        inputPanel.add(nameField);
        inputPanel.add(new JLabel("Email:"));
        inputPanel.add(emailField);
        inputPanel.add(saveButton);



        add(inputPanel, BorderLayout.NORTH);


        tableModel = new DefaultTableModel(new String[]{"ID",
"Name", "Email"}, 0);
        userTable = new JTable(tableModel);
        add(new JScrollPane(userTable), BorderLayout.CENTER);
        loadUserData();

        saveButton.addActionListener(e -> saveUser());
```

```
        // updateButton.addActionListener(e -> updateUser());
        // deleteButton.addActionListener(e -> deleteUser());

        setVisible(true);
    }

        private void loadUserData() {
        tableModel.setRowCount(0);
        try (Connection conn = DriverManager.getConnection(URL, USER, PASSWORD)) {
            String sql = "SELECT * FROM users";
            PreparedStatement stmt = conn.prepareStatement(sql);
            ResultSet rs = stmt.executeQuery();
            while (rs.next()) {
                tableModel.addRow(new Object[]{rs.getInt("id"), rs.getString("name"), rs.getString("email")});
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    private void saveUser() {
        if (idField.getText().isEmpty() || nameField.getText().isEmpty() || emailField.getText().isEmpty()) {
            JOptionPane.showMessageDialog(this, "All fields are required!", "Error", JOptionPane.ERROR_MESSAGE);
            return;
        }
        try (Connection conn = DriverManager.getConnection(URL,
USER, PASSWORD)) {
            String sql = "INSERT INTO users (id, name, email) VALUES
(?, ? ,?)";
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setString(1,idField.getText());
            stmt.setString(2, nameField.getText());
            stmt.setString(3, emailField.getText());
            stmt.executeUpdate();
            JOptionPane.showMessageDialog(this, "User Saved!");
            loadUserData();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }


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

}
```

7. CRUD operations using JDBC demo

```
package jdbcuserdemo;
import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.sql.*;

/**
 *
 * @author preety tilwani
 */
public class JDBCUserDemo extends JFrame{

    private JTextField nameField, emailField,idField;
    private JButton saveButton,updateButton,deleteButton;
    private JTable userTable;
    private DefaultTableModel tableModel;

    private static final String URL = "jdbc:mysql://localhost:3306/userdb";
    private static final String USER = "root";
    private static final String PASSWORD = "";

    public JDBCUserDemo() {
        setTitle("User Management System");
        setSize(500, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLayout(new BorderLayout());

        JPanel inputPanel = new JPanel(new GridLayout(5, 1, 10, 10));
        idField = new JTextField();
        nameField = new JTextField();
        emailField = new JTextField();
        saveButton = new JButton("Save");
        updateButton = new JButton("Update");
        deleteButton = new JButton("Delete");

        inputPanel.add(new JLabel("id:"));
        inputPanel.add(idField);
        inputPanel.add(new JLabel("Name:"));
        inputPanel.add(nameField);
        inputPanel.add(new JLabel("Email:"));
        inputPanel.add(emailField);
        inputPanel.add(saveButton);
        inputPanel.add(updateButton);
        inputPanel.add(deleteButton);


        add(inputPanel, BorderLayout.NORTH);

        tableModel = new DefaultTableModel(new String[]{"ID", "Name", "Email"},
0);
        userTable = new JTable(tableModel);
        add(new JScrollPane(userTable), BorderLayout.CENTER);
        loadUserData();

        saveButton.addActionListener(e -> saveUser());
    updateButton.addActionListener(e -> updateUser());
    deleteButton.addActionListener(e -> deleteUser());

    setVisible(true);
  }

    private void loadUserData() {
    tableModel.setRowCount(0);
    try (Connection conn = DriverManager.getConnection(URL, USER,
PASSWORD)) {
      String sql = "SELECT * FROM users";
      PreparedStatement stmt = conn.prepareStatement(sql);
      ResultSet rs = stmt.executeQuery();
      while (rs.next()) {
        tableModel.addRow(new Object[]{rs.getInt("id"), rs.getString("name"),
rs.getString("email")});
      }
    } catch (SQLException e) {
      e.printStackTrace();
    }
  }

  private void saveUser() {
    if (idField.getText().isEmpty() || nameField.getText().isEmpty() ||
emailField.getText().isEmpty()) {
      JOptionPane.showMessageDialog(this, "All fields are required!",
"Error", JOptionPane.ERROR_MESSAGE);
      return;
    }
    try (Connection conn = DriverManager.getConnection(URL, USER,
PASSWORD)) {
      String sql = "INSERT INTO users (id, name, email) VALUES (?, ? ,?)";
      PreparedStatement stmt = conn.prepareStatement(sql);
      stmt.setString(1,idField.getText());
      stmt.setString(2, nameField.getText());
      stmt.setString(3, emailField.getText());
      stmt.executeUpdate();
      JOptionPane.showMessageDialog(this, "User Saved!");
      loadUserData();
    } catch (SQLException e) {
      e.printStackTrace();
    }
  }


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

  private void updateUser() {
    String id = idField.getText();
```

```
        String name = nameField.getText();
        String email = emailField.getText();

        if (id.isEmpty() || name.isEmpty() || email.isEmpty()) {
            JOptionPane.showMessageDialog(this, "All fields are required!",
"Error", JOptionPane.ERROR_MESSAGE);
            return;
        }

        try (Connection conn = DriverManager.getConnection(URL, USER,
PASSWORD)) {
            String sql = "UPDATE users SET name = ?, email = ? WHERE id = ?";
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setString(1, name);
            stmt.setString(2, email);
            stmt.setInt(3, Integer.parseInt(id));
            stmt.executeUpdate();

            JOptionPane.showMessageDialog(this, "Data updated successfully!",
"Success", JOptionPane.INFORMATION_MESSAGE);
            loadUserData();
            idField.setText("");
            nameField.setText("");
            emailField.setText("");
        } catch (SQLException ex) {
            ex.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage(),
"Database Error", JOptionPane.ERROR_MESSAGE);
        }
    }

    private void deleteUser() {
         String id = idField.getText();

        if (id.isEmpty()) {
            JOptionPane.showMessageDialog(this, "ID field is required!", "Error",
JOptionPane.ERROR_MESSAGE);
            return;
        }

        try (Connection conn = DriverManager.getConnection(URL, USER,
PASSWORD)) {
            String sql = "DELETE FROM users WHERE id = ?";
            PreparedStatement stmt = conn.prepareStatement(sql);
            stmt.setInt(1, Integer.parseInt(id));
            stmt.executeUpdate();

            JOptionPane.showMessageDialog(this, "Data deleted successfully!",
"Success", JOptionPane.INFORMATION_MESSAGE);
            idField.setText("");
            nameField.setText("");
            emailField.setText("");
            loadUserData();
        } catch (SQLException ex) {
            ex.printStackTrace();
            JOptionPane.showMessageDialog(this, "Error: " + ex.getMessage(),
"Database Error", JOptionPane.ERROR_MESSAGE);
        }
    }

}
```