

OBE IMPLEMENTATION: PROGRAMS SETTING

by

Likkith Reddy Chirasani [AP22110010179]

Surya Teja Mannava [AP22110010154]

Abhilash Thota [AP22110010140]

Vignesh Maddela [AP22110010185]

Manoj Chandu [AP22110010085]

Akbar Sherif [AP22110010113]

A report for the CS307:Mobile Application Development using JAVA



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SRM UNIVERSITY AP::AMARAVATI

INDEX

Introduction	2
Project Modules:	3
Architecture Diagram	3
Module Description	4
Programming Details naming conventions to be used:	5
Table details:(eg university)[you consider your module]	5
Source Code	5
Screen Shots	6
Conclusion	7

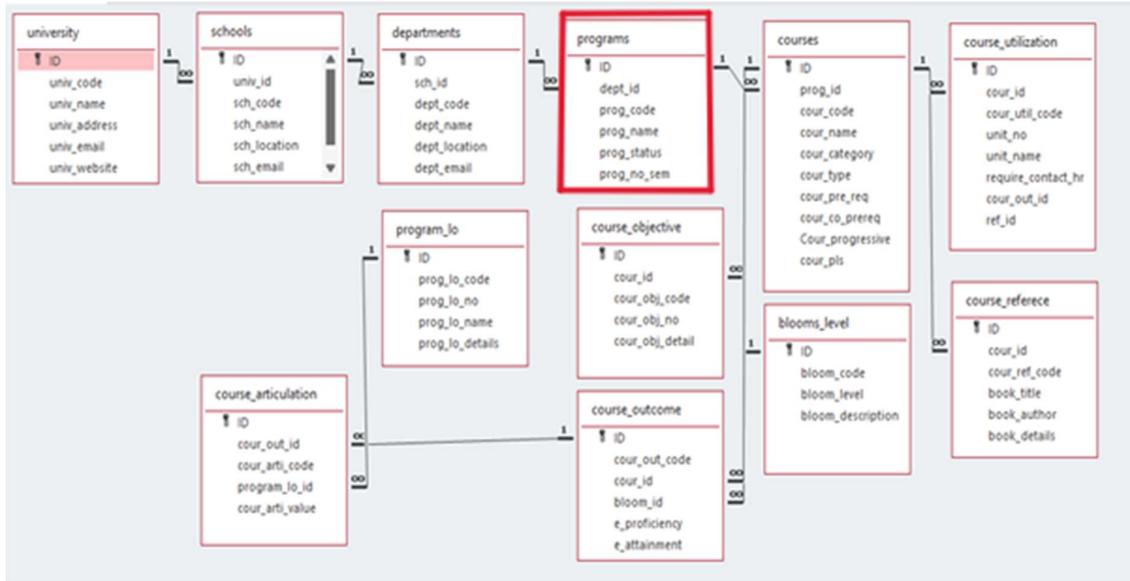
Introduction

Our University (herewith considered as SRM-AP) is going to implement OBE(Outcome Based Education) in their university and you are assigned in the project to develop a CURD(Create,Update,Retrieve and Delete) windows and mobile application using JAVA programming and Android studio for the same.

Project Modules:

- Various Modules available in the project are
- 1.Blooms Level setting
 - 2.Program Level Objective Setting
 - 3.University
 - 4.Schools
 - 5.Department
 - 6.Programs
 - 7.Courses
 - 8.Course objective setting
 - 9.Course Outcome Setting
 - 10.Course Articulation matrix Setting
 - 11.Course Utilization Setting
 - 12.Course Reference Setting.

Architecture Diagram



Module Description

Module Name: Programs

Module Description:

This module is used to create, Update, Retrieve, Delete(hereafter known as CURD) details of the module and storing the details in the database table(eg.MySQL).

Programming Details naming conventions to be used:

- **class name/activity name:** Legends_Programs
- **Function/method name**
 - **Create:** AP22110010140_Programs_create
 - **Update:** AP22110010179_Programs_update
 - **Retrieve:** AP22110010154_Programs_retrieve
 - **Delete:** AP22110010185_Programs_delete
 - **GUI:** AP22110010085 and AP22110010113

Table details:

Field Name	Data type
id	integer
dept_id	integer
prog_code	String
porg_name	String
prog_status	String
prog_no_sem	integer

Source Code

```

import javax.swing.*;
import javax.swing.table.DefaultTableModel;
import java.awt.*;
import java.sql.*;

public class Legends_Programs {
    private JFrame frame;
    private JTable table;
    private DefaultTableModel model;
    private Connection conn;

    public Legends_Programs() {
        connectDB();
        createGUI();
        AP22110010154_Programs_retrieve(); // Load table data
    }

    // Connect to SQLite DB
    private void connectDB() {
        try {
            String dbPath =
"jdbc:sqlite:C:\\\\Users\\\\likki\\\\OneDrive\\\\Desktop\\\\Programs\\\\programs.db";
            conn = DriverManager.getConnection(dbPath);
            System.out.println("☑ Connected to the database successfully!");
        } catch (SQLException e) {
            JOptionPane.showMessageDialog(frame, "☒ Database Connection Failed!", "Error", JOptionPane.ERROR_MESSAGE);
            e.printStackTrace();
        }
    }

    // Create GUI
    private void createGUI() {

```

```

        frame = new JFrame("Programs CRUD");
        frame.setSize(800, 550);
        frame.setLayout(new BorderLayout());
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel buttonPanel = new JPanel(new FlowLayout(FlowLayout.CENTER, 20,
10));
        JButton btnAdd = new JButton("✚ Add");
        JButton btnUpdate = new JButton("✎ Update");
        JButton btnDelete = new JButton("☒ Delete");
        JButton btnRefresh = new JButton("⟳ Refresh");

        buttonPanel.add(btnAdd);
        buttonPanel.add(btnUpdate);
        buttonPanel.add(btnDelete);
        buttonPanel.add(btnRefresh);
        frame.add(buttonPanel, BorderLayout.NORTH);

        model = new DefaultTableModel(new String[]{"ID", "Dept ID", "Program
Code", "Program Name", "Program Status", "Which Semester"}, 0);
        table = new JTable(model);
        JScrollPane scrollPane = new JScrollPane(table);
        frame.add(scrollPane, BorderLayout.CENTER);

        btnAdd.addActionListener(e -> openAddDialog());
        btnUpdate.addActionListener(e -> AP22110010179_Programs_update());
        btnDelete.addActionListener(e -> AP221100101185_Programs_delete());
        btnRefresh.addActionListener(e -> AP22110010154_Programs_retrieve());

        frame.setVisible(true);
    }

    // Retrieve (Load) data into table
    private void AP22110010154_Programs_retrieve() {
        model.setRowCount(0);
        try (Statement stmt = conn.createStatement(); ResultSet rs =
stmt.executeQuery("SELECT * FROM programs")) {
            while (rs.next()) {
                model.addRow(new Object[]{
                    rs.getInt("ID"), rs.getInt("dept_id"),
                    rs.getString("prog_code"),
                    rs.getString("prog_name"),
                    rs.getString("prog_status"), rs.getInt("prog_no_sem")
                });
            }
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

    // Create new program dialog

```

```

private void openAddDialog() {
    JDialog dialog = new JDialog(frame, "✚ Add New Program", true);
    dialog.setSize(400, 320);
    dialog.setLocationRelativeTo(frame);
    dialog.setLayout(new BorderLayout());

    JPanel formPanel = new JPanel(new GridLayout(6, 2, 10, 10));
    formPanel.setBorder(BorderFactory.createEmptyBorder(20, 20, 10, 20));

    JLabel deptLabel = new JLabel("Department ID:");
    JTextField deptField = new JTextField();

    JLabel codeLabel = new JLabel("Program Code:");
    JTextField codeField = new JTextField();

    JLabel nameLabel = new JLabel("Program Name:");
    JTextField nameField = new JTextField();

    JLabel statusLabel = new JLabel("Status (Active/Inactive):");
    JTextField statusField = new JTextField();

    JLabel semLabel = new JLabel("Semester:");
    JComboBox<Integer> semCombo = new JComboBox<>();
    for (int i = 1; i <= 8; i++) semCombo.addItem(i);

    formPanel.add(deptLabel); formPanel.add(deptField);
    formPanel.add(codeLabel); formPanel.add(codeField);
    formPanel.add(nameLabel); formPanel.add(nameField);
    formPanel.add(statusLabel); formPanel.add(statusField);
    formPanel.add(semLabel); formPanel.add(semCombo);

    JButton saveBtn = new JButton("💾 Save");
    JPanel bottomPanel = new JPanel();
    bottomPanel.add(saveBtn);

    dialog.add(formPanel, BorderLayout.CENTER);
    dialog.add(bottomPanel, BorderLayout.SOUTH);

    saveBtn.addActionListener(e -> {
        try {
            int deptId = Integer.parseInt(deptField.getText().trim());
            String progCode = codeField.getText().trim();
            String progName = nameField.getText().trim();
            String progStatus = statusField.getText().trim();
            int progNoSem = (int) semCombo.getSelectedItem();

            if (progCode.isEmpty() || progName.isEmpty() ||
            progStatus.isEmpty()) {
                JOptionPane.showMessageDialog(dialog, "⚠ All fields must
be filled!", "Warning", JOptionPane.WARNING_MESSAGE);
                return;
            }
        } catch (NumberFormatException ex) {
            JOptionPane.showMessageDialog(dialog, "⚠ Invalid input for
Semester field", "Error", JOptionPane.ERROR_MESSAGE);
        }
    });
}

```

```

        }

        AP221100101140_Programs_create(deptId, progCode, progName,
progStatus, progNoSem);
        JOptionPane.showMessageDialog(dialog, "☑ Program Added
Successfully!");
        dialog.dispose();
        AP22110010154_Programs_retrieve();
    } catch (Exception ex) {
        JOptionPane.showMessageDialog(dialog, "☒ Invalid Input!",
"Error", JOptionPane.ERROR_MESSAGE);
    }
});

dialog.setVisible(true);
}

// Create
private void AP221100101140_Programs_create(int deptId, String progCode,
String progName, String progStatus, int progNoSem) {
    try (PreparedStatement ps = conn.prepareStatement(
        "INSERT INTO programs (dept_id, prog_code, prog_name,
prog_status, prog_no_sem) VALUES (?, ?, ?, ?, ?)")) {
        ps.setInt(1, deptId);
        ps.setString(2, progCode);
        ps.setString(3, progName);
        ps.setString(4, progStatus);
        ps.setInt(5, progNoSem);
        ps.executeUpdate();
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(frame, "☒ Error inserting
program.");
        e.printStackTrace();
    }
}

// Update
private void AP22110010179_Programs_update() {
    int row = table.getSelectedRow();
    if (row == -1) {
        JOptionPane.showMessageDialog(frame, "☒ Select a program to
update.");
        return;
    }

    int id = (int) table.getValueAt(row, 0);
    JTextField deptField = new JTextField(table.getValueAt(row,
1).toString());
    JTextField codeField = new JTextField((String) table.getValueAt(row,
2));
}

```

```

        JTextField nameField = new JTextField((String) table.getValueAt(row,
3));
        JTextField statusField = new JTextField((String) table.getValueAt(row,
4));
        JComboBox<Integer> semCombo = new JComboBox<>();
        for (int i = 1; i <= 8; i++) semCombo.addItem(i);
        semCombo.setSelectedItem(table.getValueAt(row, 5));

        JPanel panel = new JPanel(new GridLayout(6, 2, 5, 5));
        panel.add(new JLabel("Dept ID:")); panel.add(deptField);
        panel.add(new JLabel("Code:")); panel.add(codeField);
        panel.add(new JLabel("Name:")); panel.add(nameField);
        panel.add(new JLabel("Status:")); panel.add(statusField);
        panel.add(new JLabel("Semester:")); panel.add(semCombo);

        int result = JOptionPane.showConfirmDialog(frame, panel, "Update
Program", JOptionPane.OK_CANCEL_OPTION);

        if (result == JOptionPane.OK_OPTION) {
            try (PreparedStatement ps = conn.prepareStatement(
                "UPDATE programs SET dept_id=?, prog_code=?, prog_name=?,
prog_status=?, prog_no_sem=? WHERE ID=?")) {
                ps.setInt(1, Integer.parseInt(deptField.getText().trim()));
                ps.setString(2, codeField.getText().trim());
                ps.setString(3, nameField.getText().trim());
                ps.setString(4, statusField.getText().trim());
                ps.setInt(5, (int) semCombo.getSelectedItem());
                ps.setInt(6, id);
                ps.executeUpdate();
                JOptionPane.showMessageDialog(frame, "☑ Program Updated!");
                AP22110010154_Programs_retrieve();
            } catch (Exception ex) {
                JOptionPane.showMessageDialog(frame, "☒ Error updating
program.");
            }
        }
    }

    // Delete
    private void AP221100101185_Programs_delete() {
        int row = table.getSelectedRow();
        if (row == -1) {
            JOptionPane.showMessageDialog(frame, "☒ Select a row to
delete.");
            return;
        }

        int id = (int) table.getValueAt(row, 0);
        int confirm = JOptionPane.showConfirmDialog(frame, "Are you sure?",
"Delete Program", JOptionPane.YES_NO_OPTION);
        if (confirm == JOptionPane.YES_OPTION) {

```

```

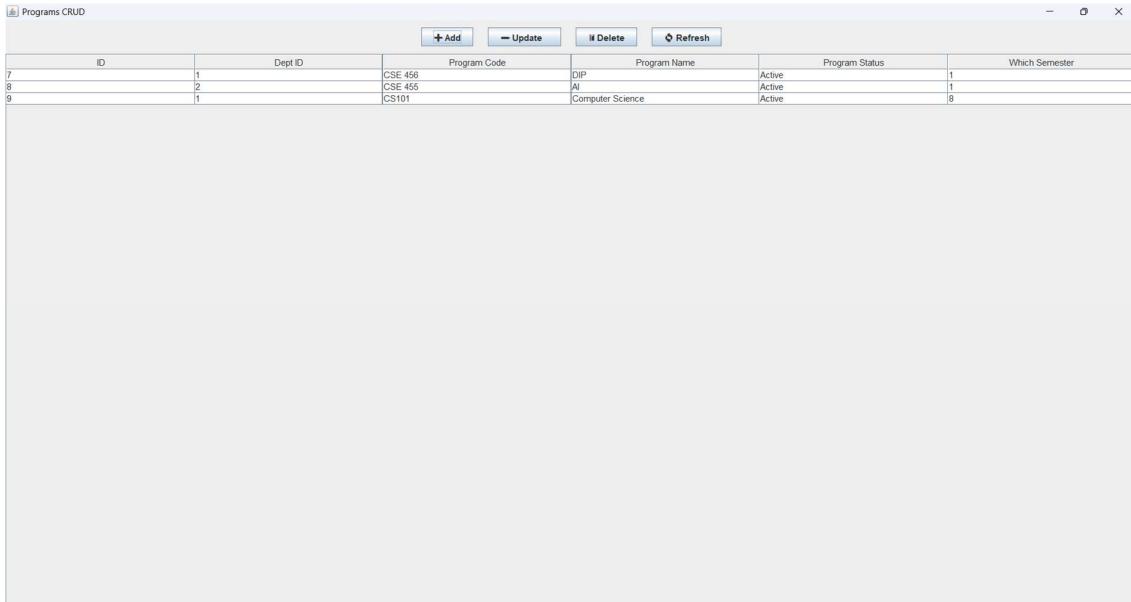
        try (PreparedStatement ps = conn.prepareStatement("DELETE FROM
programs WHERE ID=?")) {
            ps.setInt(1, id);
            ps.executeUpdate();
            JOptionPane.showMessageDialog(frame, "☑ Program Deleted!");
            AP22110010154_Programs_retrieve();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }

// Launch
public static void main(String[] args) {
    SwingUtilities.invokeLater(() -> new Legends_Programs());
}

```

Screen Shots

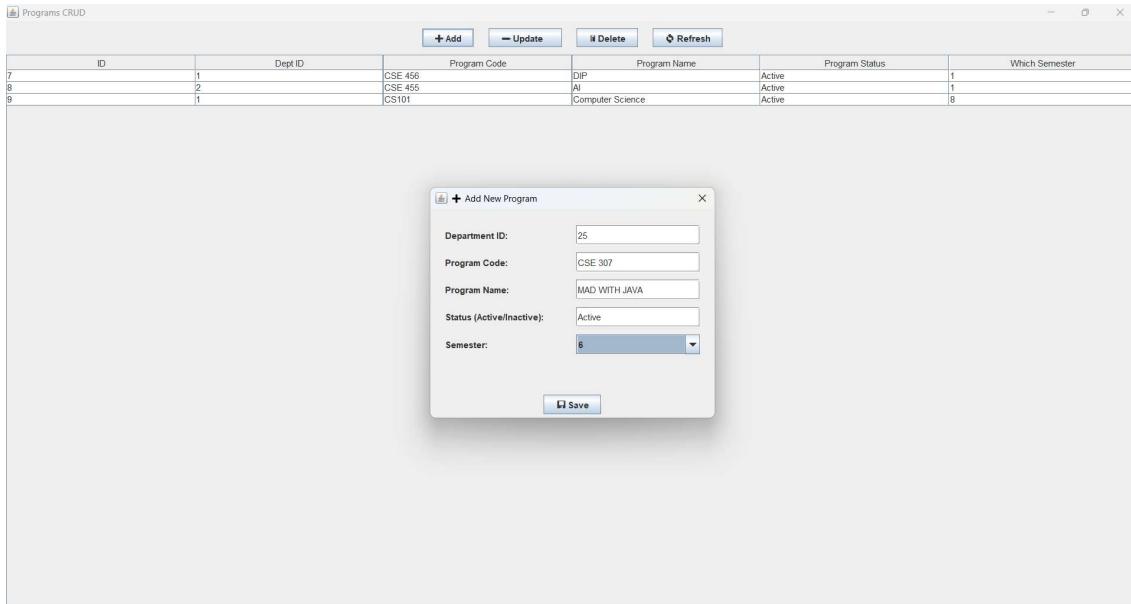
Retrieving all the Programs data from database:



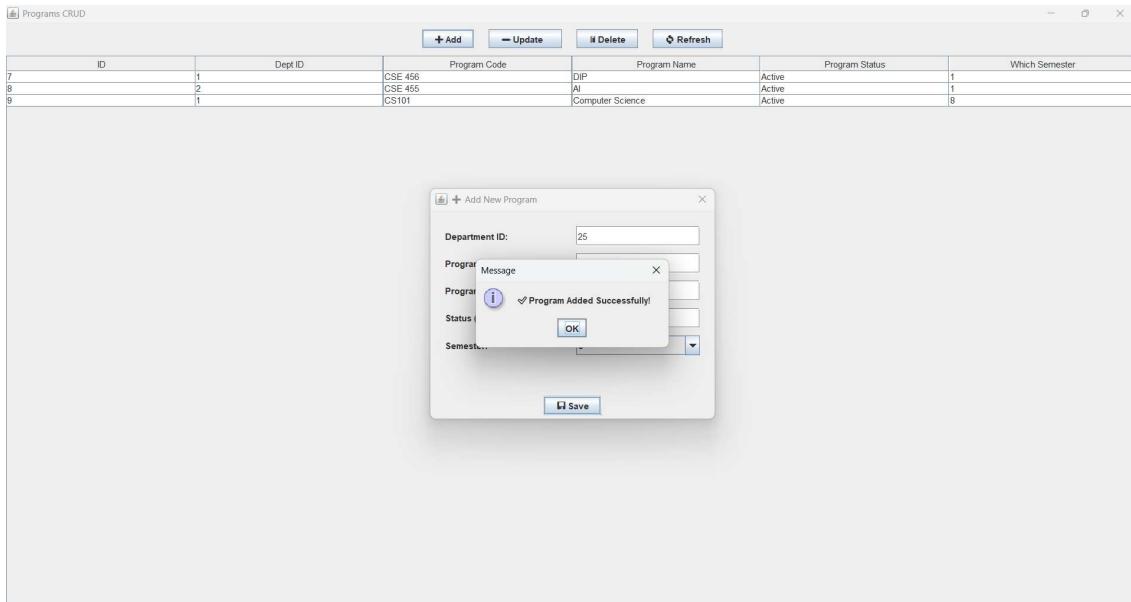
The screenshot shows a Java Swing application window titled "Programs CRUD". At the top, there is a toolbar with four buttons: "+ Add", "- Update", "Delete", and "Refresh". Below the toolbar is a table with the following data:

ID	Dept ID	Program Code	Program Name	Program Status	Which Semester
7	1	CSE 456	DIP	Active	1
8	2	CSE 455	AI	Active	1
9	1	CS101	Computer Science	Active	8

Adding a Program:



Program Added Successfully:



New Program added to the database as shown below:

ID	Dept ID	Program Code	Program Name	Program Status	Which Semester
7	1	CSE 456	DIP	Active	1
8	2	CSE 455	AI	Active	1
9	1	CS101	Computer Science	Active	8
17	25	CSE 307	MAD WITH JAVA	Active	6

Updating a Program:

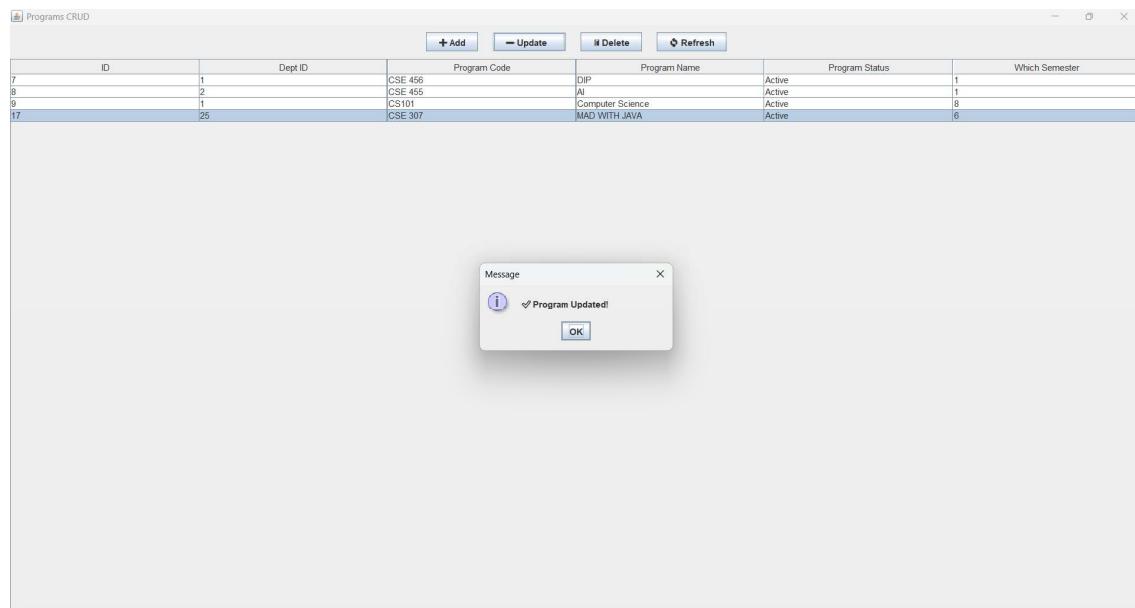
ID	Dept ID	Program Code	Program Name	Program Status	Which Semester
7	1	CSE 456	DIP	Active	1
8	2	CSE 455	AI	Active	1
9	1	CS101	Computer Science	Active	8
17	25	CSE 307	MAD WITH JAVA	Active	6

Update Program

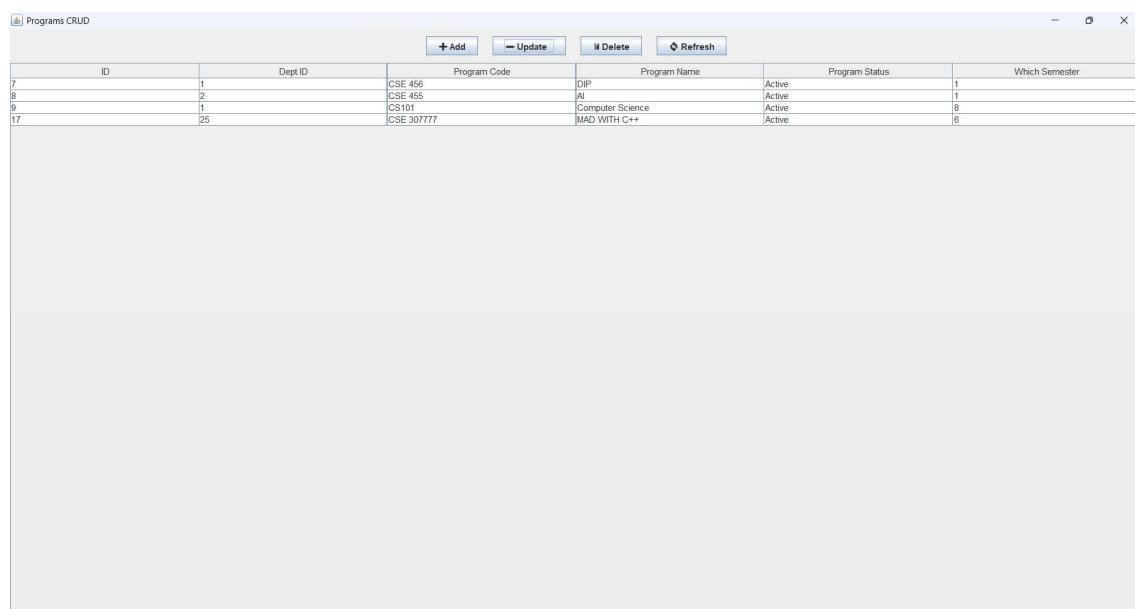
Dept ID: 25
Code: CSE 307777
Name: MAD WITH C++
Status: Active
Semester: 6

OK Cancel

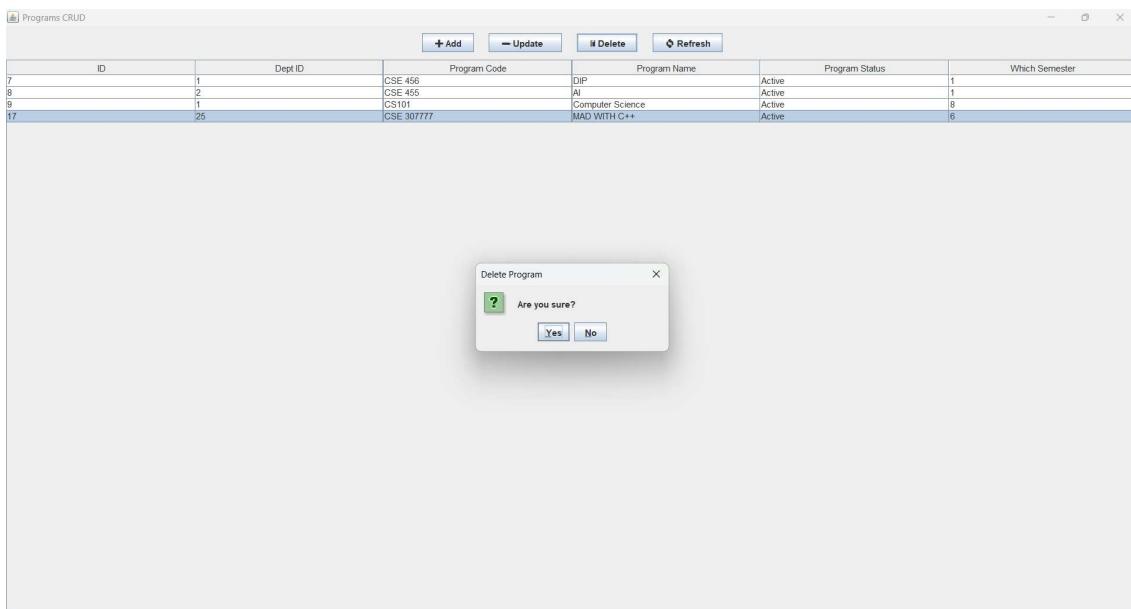
Program updated Successfully :



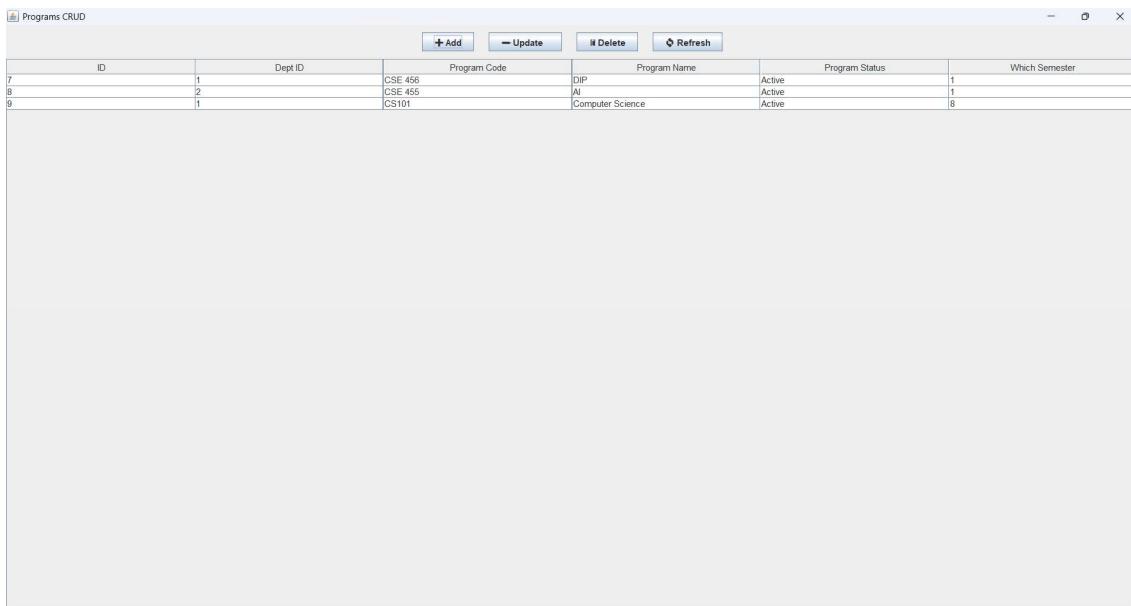
Updated changes is shown below:



Deleting a Program:



Program deleted successfully in database as shown below:



Conclusion

The *Programs CRUD Management System* has been successfully developed as a robust and user-friendly desktop application using Java Swing and SQLite. It enables efficient management of academic programs by providing core functionalities such as creating, reading, updating, and deleting program records.

This project demonstrates practical implementation of GUI design, database connectivity using JDBC, and structured modular coding practices. By integrating intuitive user interfaces with backend operations, the system offers a seamless experience for administrators to manage program data with accuracy and ease.